Iterative Reconstruction Technique for Cone-beam Computed Tomography with Limited Data

Lohvithee, Manasavee

Award date:
2019

Awarding institution:
University of Bath

Iterative Reconstruction Technique for Cone-beam Computed Tomography with Limited Data

submitted by

Manasavee Lohvithee

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Electrical and Electronic Engineering

October 2018

Contents

List of Figures

List of Tables

Acknowledgements

First and foremost, I would like to take this opportunity to express my sincere gratitude to my supervisor, Professor Manuchehr Soleimani, for an invaluable guidance and support throughout the whole process of PhD. I greatly appreciate Royal Thai Government for financial support for my PhD study. Thank you very much for giving me the opportunity to pursuit my PhD. I am forever grateful to Dr Ander Biguri. You have no idea how much you have helped me all the way through. I owe more than 80% of this PhD to you and definitely there is no way I can make it without you. Of all the unlucky things I have come across while I was doing PhD, if there is only one thing I can consider myself lucky is to have a colleague and friend like yourself. It is my greatest pleasure to have worked and spent most of my PhD life with you. I am also thankful to my beloved friends in our lab: Chenning Wu, Dr Fang Li and everyone. Thank you for the friendship and inspired discussion we shared together. It is an absolute pleasure to have known you guys. Thank you Dr Wenjuan Sun and Dr Stephane Chretien from National Physical Laboratory (NPL) for the knowledge they have shared with me. You two inspire me to never stop developing myself. I would also like to thank you all my friends I have made throughout my study especially my Thai friends. I do not have enough space to name all of you here but I sincerely thank you all for the memories we have shared together.

Massive gratitude to my dearest friends in Thailand: Purada, Kantamas, Thanattha, Nicharee and Supavadee. Thank you for your encouragement and support throughout my time in England. Although we are 6,000 miles apart, you guys never made me feel like I had to face my problems alone. There is no way I can make it to finish my study without you guys. I know I can never thank you enough. Thank you Pete for always beside me through ups and downs. I am very grateful for your support throughout my whole time in England. Last but not least, I have no words to express gratitude to my Lohvithee family. You are the best and truly are the wind beneath my wings. Thank you for always believe in me since day one and give me the strength I need. My achievements are undoubtedly also yours.

Abstract

X-ray cone-beam computed tomography (CBCT) has been extensively used in various applications, especially in medical analysis and the image-guided radiation therapy (IGRT). There have been on-going attempts to reconstruct images using a reduced number of projection data, in order to reduce the amount of radiation dose delivered to patients. However, reconstruction from insufficient number of projection data leads to reconstructed image with poor quality from severe artefacts when using analytical approach, i.e. Filtered Backprojection (FBP). In this scenario, iterative algorithms can significantly improve image quality, but comes at the cost of more complicated implementation and much longer computational time. These are the main drawbacks that make iterative algorithms difficult to be applied in a real clinical usage. This thesis focusses on developing advanced iterative algorithm to overcome the problems arising from CT reconstruction using limited number of projection data. The adaptive-weighted projection-controlled steepest descent (AwPCSD) algorithm is proposed by implementing projection onto convex sets (POCS) to enforce the data and the positivity constraints and minimising adaptive-weighted total variation (AwTV) norm. Experimental results showed that the AwPCSD algorithm is able to preserve the edges of the reconstructed image better with less number of sensitive hyper-parameters to tune, when compared to the pioneering work in this field such as the adaptive-steepest-descent POCS (ASD-POCS) algorithm. This thesis also analyses sensitivity of hyper-parameters, which are important components and play critical roles in the quality of reconstruction results. These hyper-parameters control the balance between the constraints and objective function in the TV regularised algorithms. The manual tuning of TV hyper-parameters is a tedious and time-consuming process, for which there is no well-established criteria to guarantee the optimal set of hyper-parameters for a given data apart from trials-and-errors. In order to overcome this problem, this thesis demonstrates 2 hyper-parameter selection approaches, which can be used to assist hyper-parameter selection from the user-defined ranges of hyper-parameters. The 2 algorithms employ 2 approaches, the Ant Colony Optimisation (ACO) and the Hedge, to select the best set of hyper-parameters for the implementation of the AwPCSD algorithm. Although the computational times for the training of hyper-parameters using these 2 algorithms are quite long, the set of hyper-parameters is guaranteed to produce a good quality of image, without having to manually re-select the values again. In addition, it is promising from the experimental results that the set of optimal hyper-parameters obtained from the training stage can also be applied to other datasets with the same imaging context. Thus, the time and resource spent on trying to figure out the best set of hyper-parameters for the best result of CT reconstruction using the TV regularised algorithms can be drastically saved, which eventually help to alleviate the complications of implementing iterative algorithms.

Abbreviation	Definition
CT	Computed Tomography
CBCT	Cone-Beam Computed Tomography
IGRT	Image-Guided Radiation Therapy
MRI	Magnetic Resonance Imaging
PET	Positron Emission Tomography
EPIDs	Electronic Portal Imaging Devices
CS	Compressed Sensing
TV	Total Variation
FBP	Filtered Backprojection
FDK	Feldkamp, Davis and Kress
ART	Algebraic Reconstruction Technique
SIRT	Simultaneous Iterative Reconstruction Technique
SART	Simultaneous Algebraic Reconstruction Technique
OS-SART	Ordered Subsets Simultaneous Algebraic Reconstruction Technique
CGLS	Conjugate Gradient Least Squares
PCSD	Projection-Controlled Steepest Descent
AwPCSD	Adaptive-weighted Projection-Controlled Steepest Descent
POCS	Projection Onto Convex Sets
ASD-POCS	Adaptive-Steepest-Descent Projection Onto Convex Sets
AwTV	Adaptive-weighted Total Variation
AwTV-POCS	Adaptive-weighted Total Variation Projection Onto Convex Sets
ACO	Ant Colony Optimisation
RMSE	Root Mean Squared Error
NRMSE	Normalised Root Mean Squared Error
CC	Correlation Coefficient
GPU	Graphic Processing Unit
KKT	Karush Kuhn-Tucker
ADMM	Alternating Direction Method of Multipliers
FFT	Fast Fourier Transform
PSM	Projected Subgradient Method
EPTV	TV-based Edge Preserving
XCAT	Extended Cardiac-Torso
TIGRE	Tomographic Iterative GPU-based Reconstruction
ROI	Region of Interest
CCD	Charge Coupled Device
LiF	Lithium Fluoride
TLDs	Thermoluminescent dosimeters
DT	Donoho and Tanner
ALMT	Amelunxen, Lotz, McCoy and Tropp
GA	Genetic Algorithms
MOO	Multi-object Optimisation
NSGA	Nondominated Sorting Genetic Algorithm
DQS	Dose, Quality and Speed
UQI	Universal Quality Index
LASSO	Least Absolute Selection and Shrinkage Operator

Chapter 1

Introduction and motivation

X-ray radiation has long been discovered by Wilhelm C. Roentgen in 1895. Since then, it has been extensively used in a variety of applications. In the aspect of clinical analysis, the development of the first clinical CT scanner began in 1967 with Godfrey N. Hounsfield, who received a Nobel prize in 1972. The prize is also shared with Allan M. Cormack, who independently discovered some of the image reconstruction algorithms [116].

There are mainly 3 types of beam geometry in CT: parallel, fan and cone beam. Different geometry consists of different parameters to be considered. An acquisition process, as well as forward projection computation, for each beam geometry is discussed in more detail in chapter 2.

Considering the clinical usage of x-ray CT, cone beam computed tomography (CBCT) has been used in radiation therapy since the early 2000s to give an information about patient localisation. This facilitates the implementation of the image-guided radiation therapy (IGRT). The imaging in radiation therapy is important to identify exact shape of the tumour inside the patient, in order to deliver very high energy photons to only damage the cancerous cells and spare the normal surrounding cells as much as possible. The imaging in radiation therapy is required in 2 stages, the planning and treatment stages. The planning stage requires the exact information about the electron density of the tissues that each x-ray beam needs to cross to arrive at the target cells, in order to plan about the dose delivered to the patient and the overall expected tissue damage. In the treatment stage, the information about the physiological changes of the patient, as well as the tumour, is required during each session of the treatment. This information is used to track the progress of the treatment and the amount of real dose delivered to the patient in comparison with the planning stage. All the changes are taken into account and the next sessions of the treatment can be re-planned, if needed. Although some other imaging modalities recently start to enter the play such as magnetic resonance imaging (MRI) and positron emission tomography (PET), the functionality of each one is different which can be used to compliment the CBCT, but not to completely replace it. The role of CBCT in the radiation therapy are still very strong in the present and future of radiation therapy.

Before the use of CBCT imaging, a traditional 2D imaging of an area being irradiated was formed by using electronic portal imaging devices (EPIDs). This modality of image is called

portal imaging. Not only containing information regarding patient anatomy, portal images provide information on patient dosimetry. Thus, they can potentially be used for both dosimetric and geometric verification of radiotherapy treatments [76]. Since the development of CBCT in 1990s and the utilisation of CBCT in radiation therapy in the early 2000s, the IGRT routine technique has a rapid growth worldwide. This leads to an increase of frequency of CBCT imaging, often in a daily manner as opposed to the traditional weekly portal imaging [1]. This becomes an issue of concern for a radiation dose delivered to a patient during clinical exams. In the treatment of cancer with radiation therapy, the radiation fractionation is utilised where the full dose of radiation is divided into a number of smaller doses called fractions. By doing so, healthy cells can be recovered between treatments and there are fewer toxic effects on healthy cells. Typically, a fractionation scheme divides the dose into 30 units delivered every weekday over the course of six weeks [67]. It has been reported that a high radiation dose can increase lifetime risk of cancer in patients [12] [43]. Over the course of treatment, it is possible that the radiation dose from CBCT imaging delivered to patient equal to one therapy dose fraction.

There have been numerous attempts to reduce the radiation dose in the literature. Taking the imaging dose into consideration in the planning process can reduce the overall dose delivered to patient [91], [2]. The manufacturers also tried to reduce imaging dose for the newer CBCT systems by using collimator cassettes and field size-defining blades, to limit the range of x-ray beam [1]. Many hardware-based optimal data acquisition protocols are also used for the dose reduction [87], [73]. In addition, lowering the x-ray exposure and x-ray tube voltage in CT data acquisition process are also methods to reduce the radiation dose of x-ray CT imaging [113]. However, by doing these, the data noise and the data inconsistency associated with the sparsity will be increased [125], [80].

Another way to reduce the CT radiation dose is to reconstruct an image from sparse-view projection data, i.e. reduced number of projection views per rotation around the object [22]. The idea of accurate reconstruction using a limited number of measurements came from the discovery of compressed sensing (CS) approach proposed by Donoho [35] and Candes et al [14], [17]. The results of CS lead to new approaches to signal processing and more efficient sampling schemes with fewer projections than the Nyquist sampling theorem, which was previously understood to be a necessary condition for an accurate reconstruction [15]. CS is a technique to reconstruct a signal that is known to be sparse or compressible. Most of the CT images as represented by the x-ray attenuation coefficients are sparse, especially their gradient images which can be computed as total variation (TV) norm of an image. Therefore, among the possible candidates for CS penalty, the assumption of gradient image sparseness on the image function is the most suitable one for CT imaging.

In 2006, Sidky *et al* [121] introduced the algorithm that applied the concept of the minimisation of the total variation (TV) norm of the image to perform CT image reconstruction from sparse-view projection data. The implementation of the CT reconstruction algorithm developed in this thesis is based on this principle. More details will be discussed in the chapter 4.

Insufficient amount of projection data for CT reconstruction derives from many situations: **sparse samples** is considered when the image reconstruction performed on projection data at

few views, **limited angular range** occurs when it is not possible to acquire projection data through a complete angular range, in applications such as dental radiology, surgical imaging, thoracic imaging, mammography [83] and treatment of cranial and noncranial tumours [90]. Another situation is **gaps in the projection data**, which is caused by bad detector bins. In all of these 3 examples, the projection data are not sufficient for exact reconstruction of tomographic images. Hence, it leads to incomplete data problem, which is an ill-posed inverse problem.

For a CT reconstruction using incomplete set of projection data, the main issue is based on how to estimate a tomographic image when the projection data are not theoretically sufficient for exact image reconstruction. In an attempt to solve this problem, it is important to understand the fundamental background of x-ray CT.

A CT reconstruction task can be formulated as solving a system of simultaneous linear equations:

$$\sum_{n=1}^{N} A_{mn} x_n = b_m + e, \quad m \in [1, M] \tag{1.1}$$

where $x \in \mathbb{R}^N$ is a vector representing an N dimensional image voxel in lexicographical order of size N,$b \in \mathbb{R}^M$ is the measured data from M measurements and e is the additive noise associated with the measurement. A is called the system matrix or weight matrix describing the behaviour of the x-ray that affect each single pixel in the detector. Each element of A, A_{mn}, is the weighting factor that represents the amount of influence an m^{th} ray from the source has on an n^{th} image voxel it passing through. Theoretically, there are many methods to compute the element of this matrix such as ray-voxel intersection and grid-interpolated methods. One example is to consider the intersections of each ray with the horizontal and vertical planes in the object domain and compute the length of intersections. These lengths of intersections are used as weighting factors for the weight matrix A. In chapter 2, a comprehensive study of forward projection computation using existing methods in two dimensions and three dimensions is presented in great detail. The size of matrix A in the context of CT reconstruction, especially for a standard medical imaging, is massive and is very sparse. This makes it almost impossible to store the matrix in the memory, which is one of the most common difficulties found in solving a CT reconstruction problem. A thorough understanding of the formation of matrix A is an important foundation for the development of CT reconstruction algorithm in the latter stage of this PhD.

In an ideal scenario, the image reconstruction problem is solved by finding x given a set of data b, in other words, inverting the system of linear equation 1.1. However, the system matrix A is ill-conditioned due to two main reasons: insufficient coverage in the scanning configuration or under-sampling set of projection data in the case of few-view CT scanning.

Since an exact solution for x cannot be found, the problem of reconstructing a discretised version of the image is often recast as

$$x^* = argmin_x ||Ax - b||^2 + G(x) \tag{1.2}$$

where x^* is an approximated solution. The first term of the equation 1.2 is a data fidelity term,

which minimises the discrepancy between forward projection of the image to be reconstructed and measured projection data. The l_2 norm minimisation is required due to the nature of noise in the physical x-ray model, which are Poisson and Gaussian noise. The second term, $G(x)$, is an optional regularisation term, which reflects a priori information of the desired image. The regularisation is added to reduce the space of possible solutions.

Next, we turn our attention to the background regarding CT reconstruction algorithm. In general, there are two categories of methods for CT reconstruction: analytical and iterative algorithms. The well-known technique called filtered backprojection (FBP) is the most commonly used algorithm for tomographic reconstruction [71]. This algorithm lies in the first category. As the name implied, the projection data are first filtered before it is linearly smeared back over the image points along the ray. For circular cone-beam tomography, an approximate reconstruction of a three-dimensional imaging from a set of two-dimensional projections was implemented by Feldkamp, Davis and Kress [45]. The FDK algorithm is predominantly used in clinical CT scanners and advanced commercial cone-beam scanners [101]. This method works efficiently and accurately if projection data are well sampled.

A problem with the FDK reconstruction algorithm occurs when an amount of projection data is insufficient. This makes the FDK method performs less efficiently and suffers from artifacts [121] [101], because the number of projection views does not satisfy the Shannon sampling theorem [63]. Apart from the reconstruction with limited projection data, another attempt to reduce the dose delivered to patients is to obtain CT measurements at lower doses by selecting weight-adapted protocols and/or using automatic tube current modulation systems [30], [123], [95]. In this scenario, it was reported in [105] that the iterative reconstruction technique provides similar image quality to that achievable with the FDK algorithm at 35 % less dose. Prior to their work in [105], Pontana et al also investigated the magnitude of noise reduction achievable with an iterative reconstruction techique in [106]. The results suggested that the iterative reconstruction technique significantly reduces the level of objective image noise on standard CT measurements. The iterative reconstruction technique which was used to compare with the FDK algorithm in both [106] and [105] is an iterative reconstruction technique in image space (IRIS) from Siemens Healthcare, Forchheim, Germany. In the IRIS approach, an iterative series of 3-dimensional non-linear image processing steps is performed after reconstruction of an initial high resolution image. These non-linear image processing steps are locally adapted according to the local image noise and image structure, aiming at maintaining or enhancing spatial resolution at higher object contrast and reducing noise in low contrast area without degrading the image texture. The results from these studies provide evidence that the iterative reconstruction technique is able to achieve good quality of image in the reconstruction with limited projection data and the low-dose CT measurements.

The iterative algorithms compute the final image iteratively through the loop as shown in figure 1-1. The process is repeated until the estimated image satisfies the pre-defined stopping criteria. A more detailed discussion of different iterative algorithms is presented in chapter 3.

Referring back to equation 1.2, a minimisation problem of CT reconstruction can be solved using iterative algorithms by formulating the data-consistency constraint with additional regu-

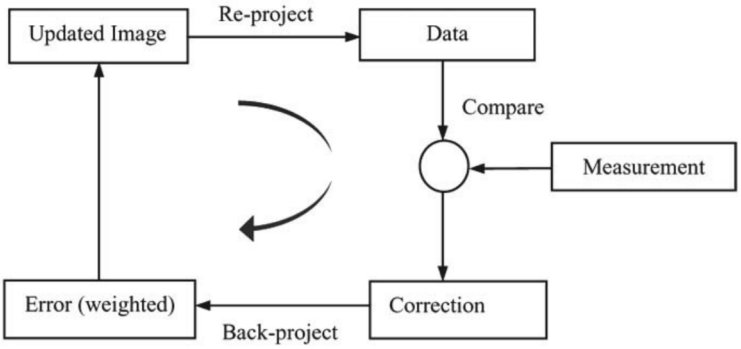


Figure 1-1: Iterative reconstruction process [98]

larisation term. There are wide range of iterative algorithms that have been developed to solve this minimisation problem. Different regularisation algorithms are discussed in chapter 4.

Despite the advantages of using iterative algorithms for solving CT reconstruction problem from limited projection data, the computational demands of iterative algorithms are much higher compared to the implementation of FDK algorithm. This is because the FDK algorithm is based on only a single reconstruction, whereas iterative algorithms use multiple repetitions to update the image until the best solution is found. The complications of iterative algorithms are the main drawbacks that make them difficult to be applied in a real clinical usage. One of the issues involves balancing the effects that the final reconstructed image has based on the data fidelity term and the regularisation term as shown in the equation 1.2. These effects are mainly governed by the values of hyper-parameters existing in these two terms. In chapter 5, the sensitivity that the reconstructed image has to value change on each hyper-parameter of some common TV regularised algorithms are analysed to understand the behaviour of all the TV hyper-parameters. The selection of appropriate values of the hyper-parameters for the TV regularised algorithms is important for the reconstruction result and the manual tuning of the hyper-parameter is a tedious and time-consuming process. Towards the end of the thesis, two hyper-parameter selection algorithms are presented to overcome the problem of selecting the optimal hyper-parameters for a given set of data.

1.1 Aim of the thesis

The aim of this thesis is to develop iterative reconstruction algorithms to produce a good quality of image from a set of limited x-ray CT projection data. Ultimately, the algorithms can be beneficial for medical applications in an attempt to reduce the CBCT radiation dose delivered to patients in the applications such as IGRT where regular CBCT scanning is required. In addition, the algorithms could also be applied to industrial applications to reduce the time of the x-ray CT scanning. The main research problems tackled in this thesis are as following:

- Explore the image reconstruction problem with focus on reconstruction using limited data from CBCT projections. The existing algorithms and their drawbacks are focussed to de-

velop new algorithms and produce good quality of image.

- Address the problems of using advanced iterative algorithms, especially the total variation (TV) regularised algorithms, in real practice. The sensitivity analysis of hyper-parameters required for implementation of TV regularised algorithms is implemented, in order to understand the behaviour of each hyper-parameter and the effects on the reconstruction results.

- The reconstruction problem of CT imaging from limited data is a challenging problem, which requires advanced algorithms such as TV regularised algorithms to solve the problem. One of the difficulties of implementing these algorithms is the selection of hyper-parameter values, for which the theory to solve the hyper-parameter selection in inverse problem literature is not extensive. The results often rely on the experienced users to select the best set of hyper-parameters for the best results, which make the algorithms difficult to implement for the general users. This thesis demonstrates 2 hyper-parameter selection algorithms that the computer can use to select the best set of hyper-parameters from the ranges of user-defined values. The aim is to alleviate the difficulty of choosing the values of hyper-parameters for the implementation of TV regularised algorithm, which can be problematic for the users who are not experts in the field.

1.2 Thesis organisation

The content of each chapter in this thesis is summarised as following:

Chapter 2: Fundamental of X-ray Computed Tomography

In this chapter, the acquisition of x-ray CT projection is presented. The highlight of this chapter is the detailed explanation of forward projection which is the major building block of the image reconstruction algorithms. This chapter is aimed at the readers who are interested in understanding how the forward projections in both 2D and 3D are formed step-by-step in terms of system geometries and programming codes.

Chapter 3: X-ray CT image reconstruction algorithms

This chapter explains the implementation of two main categories of x-ray CT reconstruction algorithms: analytical approach, which consists of Filtered backprojection (FBP) and iterative algebraic reconstruction algorithms such as Algebraic Reconstruction Technique (ART), Simultaneous Iterative Reconstruction Technique (SIRT), Simultaneous ART (SART) and Conjugate Gradient Least Squares (CGLS) algorithms. The advantages and disadvantages as well as convergence rate of these algorithms are discussed.

Chapter 4: Total variation regularisation algorithms

In this chapter, the total variation minimisation algorithms are reviewed. The main drawbacks of existing algorithms are presented, leading to the proposal of new algorithm: the adaptive-weighted projection-controlled steepest descent (AwPCSD). The AwPCSD algorithm is able to address the over-smoothing problem of the TV regularised algorithm with less sensitive hyper-parameters to tune. The selection of hyper-parameters is crucial for an efficient implementation of TV regularised algorithms. The in-depth analysis of hyper-parameters is presented in chapter 5.

Chapter 5: Sensitivity analysis of TV hyper-parameters

In this chapter, a comprehensive valuation of hyper-parameter selection in 4 common TV regularised algorithms (ASD-POCS, AwTV-POCS, PCSD and AwPCSD) is presented. Changes in values of each hyper-parameter are experimented and the corresponding effects on reconstruction results of 4 algorithms are analysed. The aim is to understand the behaviour that TV hyper-parameters have on the implementation of TV regularised algorithms, in order to know which ones to prioritise when tuning the algorithms. Ultimately, the experiments in this chapter are conducted to find out the suggested ways to select some hyper-parameters to minimise or completely avoid re-running the reconstruction with different hyper-parameters. In addition, the performance and robustness of the AwPCSD reconstruction algorithm is also evaluated by comparing with other 3 existing algorithms throughout the experiments for sensitivity analysis in this chapter.

Chapter 6:Hyper-parameter selection using Ant Colony Optimisation algorithm

This chapter presents the algorithm to automatically select the best set of hyper-parameters from the user-defined ranges of hyper-parameters using Ant Colony Optimisation (ACO) approach. The main reconstruction algorithm is the AwPCSD algorithm, which is the TV regularised algorithm proposed in chapter 4 that requires 5 hyper-parameters to be tuned for its implementation. The ACO approach, which imitates the way the colony of ants in nature find the shortest path from their nest to the food source, is adopted to find the best set of hyper-parameters for the AwPCSD algorithm. Therefore, the hyper-parameter selection algorithm presented in this chapter combines the ACO approach with the AwPCSD algorithm. For a given set of limited projection data, the algorithm is able to identify the best set of hyper-parameters for the reconstruction using the AwPCSD algorithm from the set of initial values of hyper-parameters as specified by the user at the beginning of the implementation.

Chapter 7: Efficient hyper-parameter calibration in Total Variation based CT reconstruction using Freund and Shapire Hedge algorithm

In this chapter, the second hyper-parameter selection algorithm is presented, which takes the motivation from the previously presented algorithm by considering the hyper-parameter selec-

tion problem from a new perspective. This algorithm combines the Hedge method of Freund and Shapire with the sequential reconstruction using the AwPCSD algorithm based on successive incorporation of new projections. The hyper-parameter selection problem can be considered as the experts problem, where each initial configuration of hyper-parameters is an expert. The loss after each iteration of the AwPCSD algorithm is computed as the RMSE error from the prediction of the next projection. New projection data is then added and the algorithm proceeds to the next iteration. A probability distribution on the hyper-parameter configurations is updated with every new projection. At the end of the implementation, the hyper-parameter configuration with the highest probability is considered as the best set of hyper-parameters for the reconstruction with the AwPCSD algorithm. The main result of this hyper-parameter selection algorithm is that choosing the value of hyper-parameters using the probability mass after a certain number of steps provides a prediction error which is almost as accurate as the prediction error incurred by the best predictor.

Chapter 2

Fundamentals of X-ray Computed Tomography

The term 'tomography' is referred to as the process of finding a cross-sectional imaging of an object from either transmission or reflection data collected by illuminating the object from many different directions [71]. In the context of X-ray computed tomography (CT), the main goal is to achieve accurate reconstruction of an image from its measurements or projections. An X-ray source is located on one side as shown in the X-ray imaging setting in figure 2-1 and irradiates an object in the middle through to a detector on the other side. The detector could be either an X-ray film or a digital sensor. The radiation as detected by the detector is called projection data. Then, the X-ray source and the detector move simultaneously in a circular motion to acquire data from different angles [114].

After projection data from multiple angles are acquired, they are backprojected to retrieve an image. The projection and backprojection are two important parts of CT reconstruction in its simplest form. In this chapter and the next chapter, we will investigate how the image of the cross-section of an object can be recovered from the projection data. The CT reconstruction made it possible for an internal inspection of the object without undergoing an invasive approach.

The accuracy of the reconstruction process depends highly on the forward operator. Different variants of forward operators come from different beam geometries of the CT imaging.

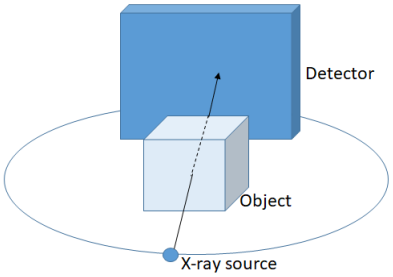


Figure 2-1: X-ray imaging setting

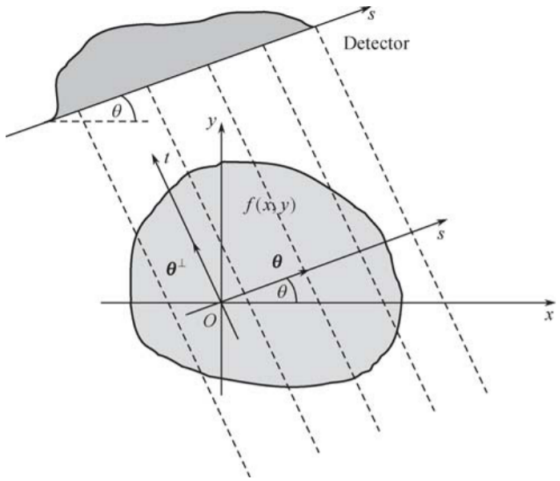


Figure 2-2: Parallel beam imaging geometry [143]

The main categories consist of parallel beam, fan beam and cone beam. This chapter presents the basic fundamental of X-ray CT. Firstly, the acquisition of X-ray CT projection is explained. Then, the computation of forward projection in 2D and 3D are presented in great details. The chapter aims specifically at the readers who are beginners of the X-ray CT field and willing to understand how the forward operators of 2D and 3D CT geometries are calculated.

2.1 Acquisition of X-ray CT projections

The most basic geometry of X-ray CT is parallel-beam geometry. A number of x-ray sources are located on one side and emanate parallel rays through the object to the detectors on the other side, as illustrated in figure 2-2.

The source-detector combination rotates through a certain angular interval and scans over the length of the next projection and so on. This is performed to collect all projections from different angles.

The acquisition process of x-ray CT projections is explained in this section. First of all, the meaning of line integrals and how they are combined to form projections of an object are explained. According to figure 2-2, the object is modelled as a density function in the $x-y$ plane, $f(x,y)$. A line integral represents the integral of some characteristic of the object along a line. In the case of x-ray CT, the characteristic being considered is attenuation of x-ray beams as they travel from x-ray sources through biological tissue. Each line integral is represented by the (θ, s) parameters. In the ideal scenario, each x-ray beam is infinitesimally very small in cross-section for which it can be considered as a straight line. In addition, the photons are assumed to be travelling along paths parallel to each other with no loss of beam intensity from beam divergence. This approximation is used for the calculation of x-ray CT projections. The equation of

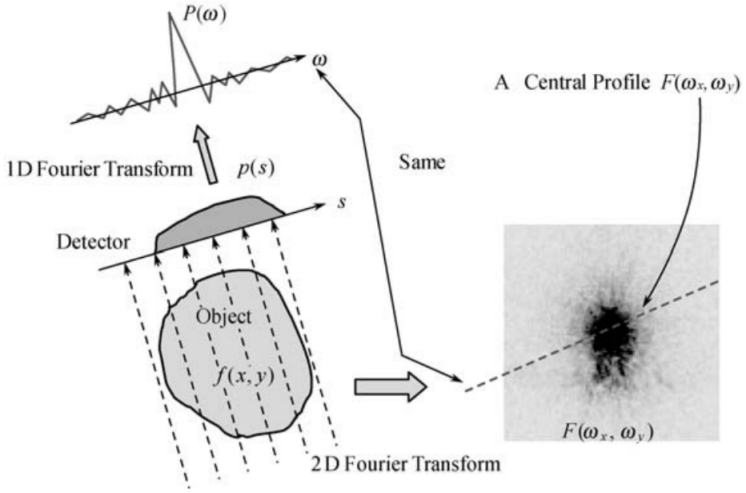


Figure 2-3: 2D central slice theorem [143]

line representing each parallel ray as shown in figure 2-2 is defined as:

$$x\cos\theta + y\sin\theta = s \tag{2.1}$$

A line integral of total attenuation suffered along one particular beam of x-ray L as it travels in a straight line through the object can be defined by the following equation [71]

$$p(s,\theta) = \int_L f(x,y)ds \tag{2.2}$$

which can be rewritten by using the Dirac delta function (δ) as:

$$p(s,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - s)dxdy \tag{2.3}$$

The equation 2.3 is the Radon transform of the object $f(x,y)$. It converts the spatial data into frequency domain. One projection view is constituted of a set of line integrals of attenuation coefficient, as measured on the detector for a given angle. The x-ray CT imaging system is then rotated to the next angles to acquire multiple projection views of the object over a range of angular orientations.

2.1.1 Central Slice Theorem

Next, the Central Slice Theorem, which is the fundamental theorem of x-ray CT is presented. This theorem, also known as the Fourier Slice Theorem, calculates the Fourier transform of a parallel projection of a 2D object function $f(x,y)$ taken at angle θ. The illustration of central slice theorem is shown in figure 2-3. The theorem states that given a projection $p(s)$ of a 2D object function $f(x,y)$ at angle θ, the 1D Fourier transform $P(\omega)$ of the projection is equivalent to a 1D profile along the radial line at the same angle of 2D Fourier transform ($F(\omega_x,\omega_y)$) of that function.

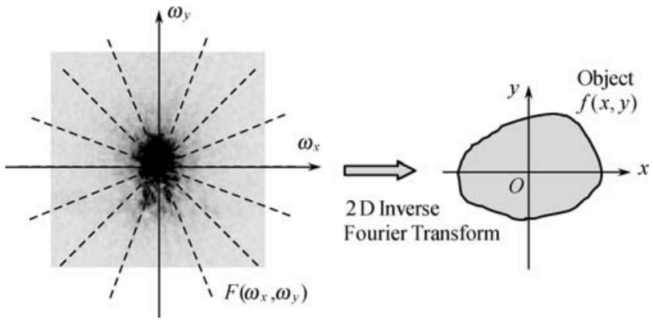


Figure 2-4: Image is reconstructed from 2D inverse Fourier transform [143]

Starting by defining the 2D Fourier transform of the object function in the polar coordinate as in the following equation

$$P(\omega,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi\omega(xcos\theta+ysin\theta)}dxdy \tag{2.4}$$

The RHS of equation 2.4 represents the 2D Fourier Transform at a spatial frequency of (ω_x = $\omega cos\theta$, ω_y = $\omega sin\theta$). Therefore, it can be derived that

$$P(\omega,\theta) = F(\omega_x,\omega_y) = F(\omega cos\theta,\omega sin\theta) \tag{2.5}$$

If the sufficient numbers of projections are taken from 0 toπ radian, the 2D Fourier transform $F(\omega_x,\omega_y)$ will be collected over the entire 2D Fourier space. Each projection from each particular angle adds a line in the Fourier space. Therefore, knowing $F(\omega_x,\omega_y)$, the original object function $f(x,y)$ can be recovered by calculating the 2D inverse Fourier transform. This process is illustrated in figure 2-4. The 2D inverse Fourier transform in the polar coordinate is defined as

$$f(x,y) = \int_0^{2\pi}\int_0^{\infty} F(\omega_x,\omega_y)e^{j2\pi\omega(xcos\theta+ysin\theta)}d\omega d\theta \tag{2.6}$$

However, there are some challenges in applying the direct Fourier space reconstruction by using the central slice theorem in actual implementation. The first challenge is due to the fact that the central slice theorem produces samples in the Fourier domain, which is not coincident with Cartesian coordinates and requires interpolation of the samples into the Cartesian coordinates. An interpolation in Fourier domain is highly sensitive, for which an interpolation of one sample affects the appearance of the entire image after the inverse Fourier transform [141]. Another challenge is the difficulties of performing targeted reconstruction to examine fine details of a small region in the object [59]. Due to these drawbacks, the alternative implementation to the central slice theorem is considered. The most commonly used algorithm is the Filtered Backprojection (FBP), which is an analytic reconstruction approach. The FBP algorithm reconstructs an image based on the principal of central slice theorem. More detail of the FBP method is presented in the chapter 3, along with other types of reconstruction algorithms.

2.2 Forward Projection of two-dimensional CT

Reconstruction of line integral measurements of x-ray CT depends heavily on the forward operators, which are the models for the acquisition process. The computation of forward projection is the major building block of the image reconstruction algorithms. It is necessary to have an understanding of how the forward projection is calculated, in order to develop x-ray CT image reconstruction algorithms.

This section explains the acquisition of 2D forward projection in details. It aims at the beginners of the CT reconstruction field to be able to understand how the forward projection is formed step-by-step, in terms of system geometries and programming codes. The accuracy of weighting factors in the system matrix A of the system of simultaneous linear equations, as expressed in equation 1.1, is very important for the final solution. The simulation of forward projection in this chapter is based on Siddon's ray-driven projection approach [119] [61], which traverses each individual ray through the voxel volume to accumulate the line-integration values. The system matrix A is constructed based on the computation of the ray-voxel intersection. Each element of matrix A contains the length of intersection between a straight path and imaging voxel multiplied by the voxel value. Different methods to compute and assign the elements of the matrix A are investigated in this chapter. The main tool used to study and implement forward projection is MATLAB, by following some existing programming codes available from different sources.

2.2.1 Parallel-beam geometry

A simple illustration of intersections between parallel rays and an object in one particular angle is shown in figure 2-5. In the this section, the way the existing software, the Algebraic Iterative Reconstruction (AIR) tools, computes the parallel-beam forward projection is first presented. Then, a slightly modified approach using a ray-casting technique is explained in details later on.

Parallel-beam forward projection from Algebraic Iterative Reconstruction (AIR) tools

The first existing software to compute parallel-beam forward projection is Algebraic Iterative Reconstruction (AIR) tools, which is the MATLAB package for Algebraic Reconstruction Techniques proposed by Per Christian Hansen and Maria Saxild-Hansen [56]. There are three tomography test problems introduced in the software package: parallel and fan beam tomography and seismic tomography. In this section, the parallel beam tomography forward projection acquisition is studied.

In order to apply the parallel beam test problem from the AIR tool software package to algebraic reconstruction methods, we need a formulation of a linear system according to the algebraic model as expressed in equation 1.1.

An object being irradiated by the parallel beam is called a domain. The user can specify the number of discretisation intervals, N, in each dimension such that the domain consists of N^2 cells. Each element of the domain is numbered from 1 to N^2 starting from the upper left corner

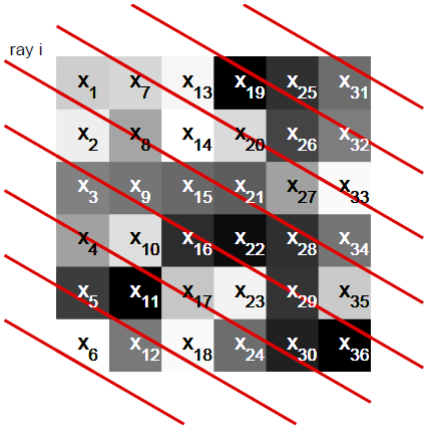


Figure 2-5: An illustration of parallel beam x-ray passing through a plane of an object containing 36 pixels (from x_1 at the top left corner to x_{36} at the bottom right corner). For a particular angle shown in the figure, there are 8 parallel rays passing different pixels of the object with different intersection lengths. A range of colours shown in the figure represent examples of different densities of components within the object. [72]

running along the columns to the lower right corner as illustrated in figure 2-5. Each element j is assigned a value x_j, which is an average of linear attenuation coefficient within the $j'th$ cell. Therefore, the vector x is a discretised version of the real linear attenuation coefficient.

Considering ray i in figure 2-5, this ray originates from a source and passes through numbers of cells in the domain in different portions. The length of intersection of the $i'th$ ray through the cell j is defined as a_{ij}. The element a_{ij} is assigned to 0, if the ray i does not pass through the cell j. The user can choose to specify the desired number of parallel rays and a range of angles θ in degrees. The sources are then rotated around the object domain in different angles θ to cover the whole object for a better representation.

Let us assume that the object domain of the system under study is divided into 50 parts of unit length in each dimension, i.e. $N = 50$. Thus, the domain consists of 50^2 cells. The range of angles is specified as 0° to 179° with 1° increment. This means that there are 180 angles being determined and number of parallel rays for each angle is 75 rays.

The distance in the domain from the first ray to the last ray is defined as:

$$d = \sqrt{2}N = \sqrt{2} \times 50 = 70.7107 \tag{2.7}$$

A multiplication of the number of discretisation intervals (N) with $\sqrt{2}$ is suggested as a default option by the AIR toolbox, in order to ensure that the whole object is within the view of the rays. The starting coordinates for all 75 parallel rays at the first angle are defined by generating 75 points which are linearly equally spaced between $-\frac{d}{2} = -35.3553$ and $\frac{d}{2} = 35.3553$, as illustrated in figure 2-6.

Fifty one constant lines are drawn within the object domain for each unit cell, starting from $x = -25$ upto $x = 25$ in x direction and $y = -25$ upto $y = 25$ in y direction. The illustration of

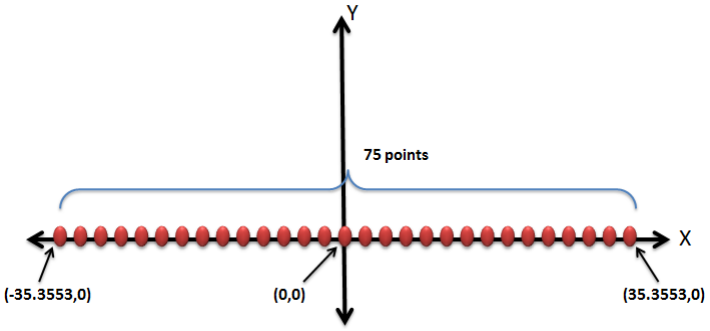


Figure 2-6: An illustration of starting coordinates for 75 parallel rays of parallel-beam forward projection using AIR tools. The figure shows starting coordinates at the first angle (0°) by generating 75 points equally spaced between -35.3553 and 35.3553 to cover the distance in the domain from the first ray to the last ray of 70.7107.

all the lines in the object domain is shown in figure 2-7.

A set of parallel rays is rotated over 180 angles around the domain. To carry out a rotation, the starting points as defined in figure 2-6 are first written as a vector and multiplied by a rotation matrix calculated from each angle θ.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.8}$$

where (x', y') are the coordinates of the points after rotation.

This is repeated over all angles. In each angle, the coordinates of the points after rotating for θ degree are used as the starting points of the rays for the current angle. The direction vectors are defined for all 75 rays corresponding to the current angle as follows.

$$a = -sin\theta \tag{2.9}$$

$$b = cos\theta \tag{2.10}$$

where a is the direction vector for x coordinates, b is the direction vector for y coordinates, and θ is the current angle. These direction vectors are used to define the parametrisation of the line to obtain the intersections between parallel rays and lines in the object domain later on.

In each one particular angle, we loop over the set of parallel rays to find the coordinates of the intersections between each ray with the object lines. An illustration of how the parallel rays intersect with the object lines in one particular angle is shown in figure 2-8.

In order to obtain the coordinates of intersections along each parallel ray, we use the parametrisation of line, as briefly explained here. According to Paul Dawkins' Online Math Notes [32], the parametric form of equation of a line can be defined, as shown in figure 2-9. This form allows us to find out any points on the lines. Suppose that an arbitrary point on the line in 3D is known to be $P_0 = (x_0, y_0, z_0)$. A vector, $\vec{v} =< a, b, c >$, is some vector that is parallel to the line.

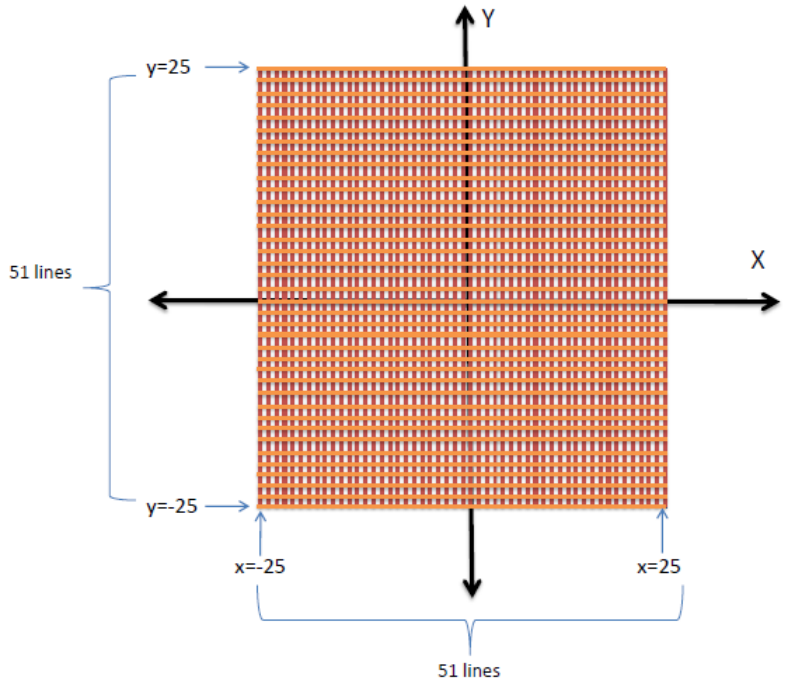


Figure 2-7: An illustration of an object domain for the parallel-beam forward projection using AIR tools. The object domain of the system under study is divided into 50 parts of unit length in each dimension. Thus, the domain consists of 50^2 cells. Fifty one constant lines are drawn in x and y dimensions, starting from $x = -25$ to $x = 25$ and $y = -25$ to $y = 25$, respectively.

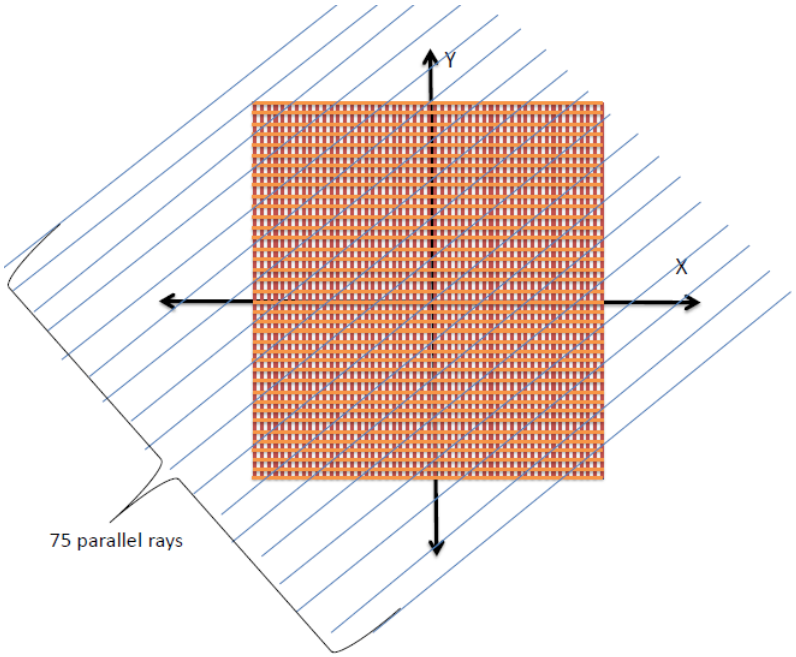


Figure 2-8: Parallel rays intersect with the object domain at one particular angle.

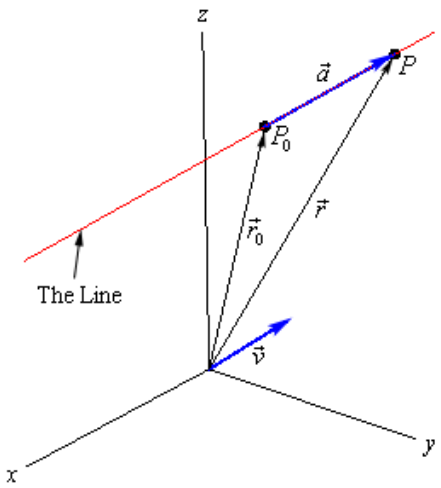


Figure 2-9: A simple sketch for the parametrisation of line [32].

The task is to find a point $P = (x, y, z)$, which is any point on the line.

Let $\vec{r_0}$ and $\vec{r}$ be position vectors for points P_0 and P, respectively. And $\vec{a}$ is a vector representing $\vec{P_0P}$. By simple vector arithmetic, we can say that $\vec{r} = \vec{r_0} + \vec{a}$ from the sketch in figure 2-9. Since the $\vec{a}$ and $\vec{v}$ are parallel to each other, it can be concluded that there is a parameter t such that $\vec{a} = t\vec{v}$.

Therefore,

$$\begin{aligned} \vec{r} &= \vec{r_0} + \vec{a} \\ &= \vec{r_0} + t\vec{v} \\ &= (x_0, y_0, z_0) + t < a, b, c > \\ < x, y, z > &= < x_0 + ta, y_0 + tb, z_0 + tc > \end{aligned}$$

The parametric form of the equation of a line can then be defined as:

$$x = x_0 + ta \tag{2.11}$$

$$y = y_0 + tb \tag{2.12}$$

$$z = z_0 + tc \tag{2.13}$$

In order to obtain coordinates of a point on the line, the parameter t is chosen and substituted into above equations 2.11,2.12,2.13. The parametric form of the line for parameter t can be defined as following:

$$t = \frac{x - x_0}{a} = \frac{y - y_0}{b} = \frac{z - z_0}{c} \tag{2.14}$$

where t_x, t_y and t_z are for x, y and z directions, respectively, x, y, z are coordinates of any points on the ray, x_0, y_0, z_0 are coordinates of the starting points of the ray and a, b, c are direction vectors. Equation 2.14 is called 'Symmetric equations of the line'.

The principle of parametrisation of line explained so far is for 3D. Since the system under determine is 2D, only two equations for x and y dimensions are needed in this case. From equation 2.14, x and y coordinates of intersections between parallel rays and object lines can be determined using the following equations:

$$x = a \times t_x + x_0 \tag{2.15}$$

$$y = b \times t_y + y_0 \tag{2.16}$$

All the coordinates of the intersections between parallel rays and object lines can be obtained by using the equations 2.15 and 2.16, looping over each angle and each ray.

Then, the software calculates the length of each parallel ray within cells by applying Euclidean distance formula:

$$d = \sqrt{dx^2 + dy^2} \tag{2.17}$$

where dx and dy are differences of x and y coordinates in an arbitrary cell that a ray passes through. These distances are substituted into a system matrix A at the corresponding x and y coordinates of the object.

Parallel-beam forward projection method using a ray-casting technique

This method is slightly modified from the previous section by using ray-casting technique to create the weight matrix. The ray-casting technique looks into each parallel ray and computes the intersections between the ray and the object.

The method to create the weight matrix using coordinates approach is introduced in [128]. Its implementation is explained in details here. First of all, the equation of line as expressed in equation 2.1 is revisited again with different notation t_k, instead of s, on the RHS:

$$xcos\theta + ysin\theta = t_k \tag{2.18}$$

The variable t_k represents a perpendicular distance of a straight line with respect to the line passing through the origin. For example, the equation 2.18 with t_k equals to 0 represents a straight line through the origin, making an angle θ to the x axis. A new line can be defined with different value of t_k. The parallel rays are simulated using this approach, as illustrated in figure 2-10.

In order to create the weight matrix using this approach, we loop over every projection angle and compute the value of t_k for each ray using equation 2.18, by substituting the x and y coordinates for each pixel of the object domain. After calculating these distances for each ray,

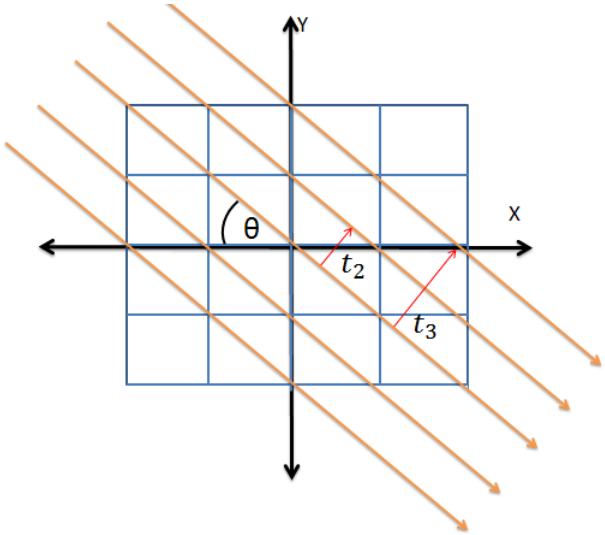


Figure 2-10: The parallel rays are simulated with different values of t_k

we can now determine which pixels of the object domain are contained in which ray. This is derived from the fact that, for one particular ray, the value of t_k calculated from the pixels of the object domain that this ray passing through should be similar. Figure 2-11 illustrates how this approach works for one particular case of 3×3 pixels domain in the projection angle of 45° .

According to figure 2-11, the coordinates of each pixel of the object domain as shown in figure 2-11(a) are substituted in equation 2.18 to calculate values of t. We can see that pixels in the same direction of one particular ray have the same values of t. Using this knowledge, the weight matrix can be constructed by adding proper value to the correct cell of the matrix.

For each projection angle, we compute the values of t in every pixels and loop over each ray to search for similar value of t. Then, the values of 1 are added to the corresponding elements of weight matrix representing the positions of these pixels. This approach makes it possible to identify which pixels in the object domain that the ray passing through and correctly assign the value to the corresponding elements of the weight matrix. The process is repeated for every parallel rays in every projection angles to achieve the complete system matrix A.

2.2.2 Fan-beam geometry

In this section, we move on to another type of 2D forward projection in CT which is called fan-beam CT. In fan-beam geometry, a point source irradiates an object with a fan-shaped x-ray beam. A bank of detectors is located on the other side to simultaneously detect all the measurements in one fan. Then, the source and the entire bank of detectors are rotated together to generate the next projections and so on [72]. Figure 2-12 compares a fan-beam imaging geometry and parallel-beam imaging geometry.

The method used to study fan-beam CT forward projection is extended from the methods described in parallel-beam section. One snapshot from the programming codes showing how the fan-beam CT intersects with the object domain at one particular angle is shown in figure

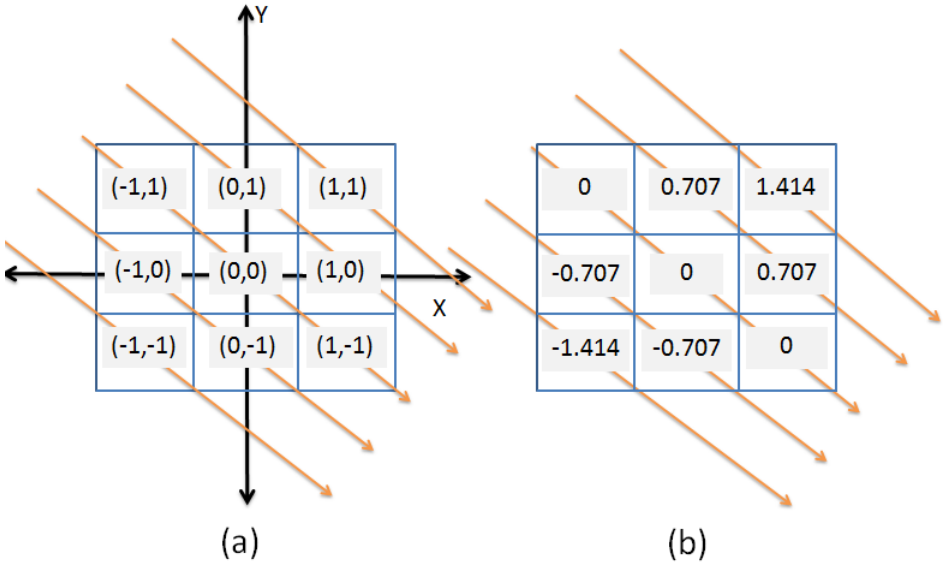


Figure 2-11: Illustration of parallel beam at one particular angle (45°); (a) Coordinates of pixels, (b) Values of t calculated using equation 2.18

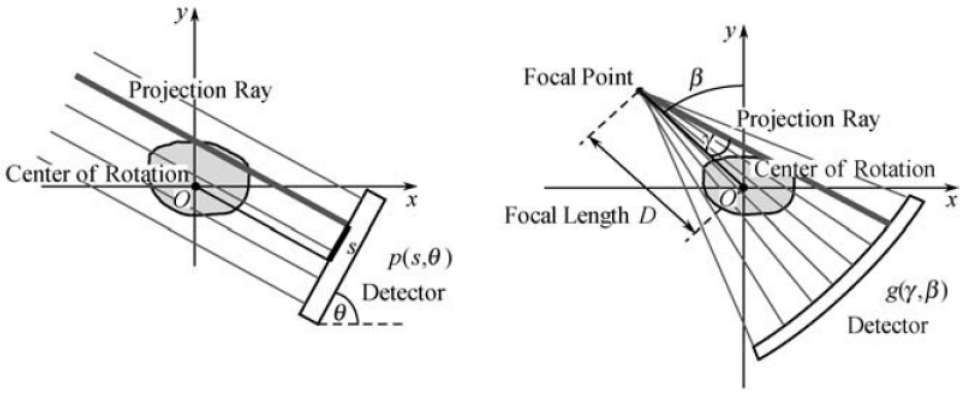


Figure 2-12: Comparison of the parallel-beam and the fan-beam imaging geometries [143]

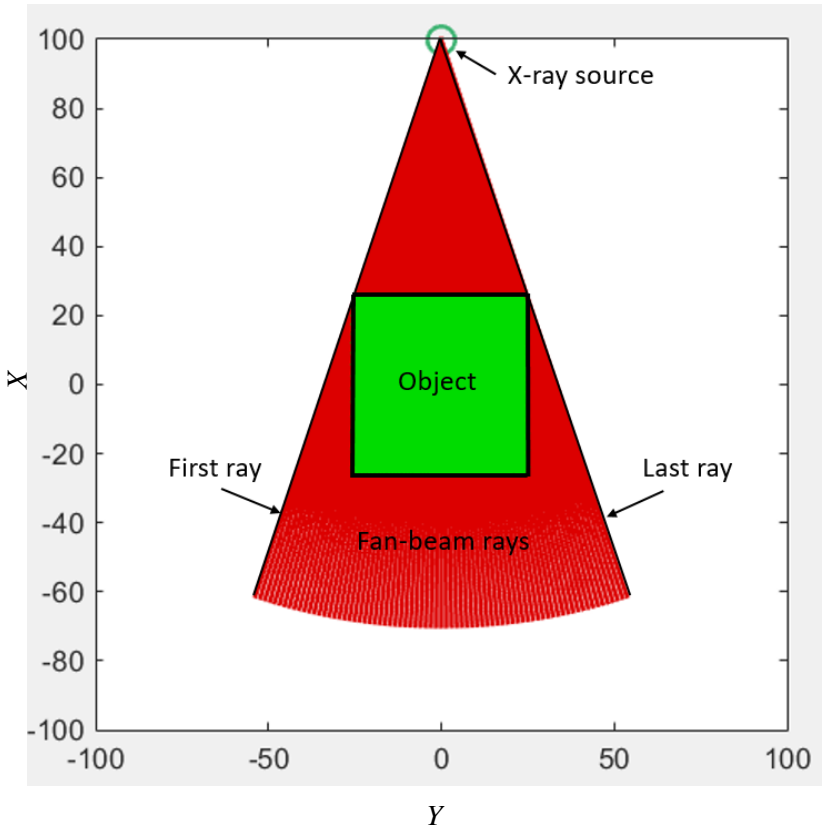


Figure 2-13: Intersection of fan-beam CT with the object domain at one particular angle.

2-13.

The weight matrix for fan-beam CT forward projection is created using the same method as a ray-casting technique for parallel-beam geometry. For each projection angle, values of t_k are computed for each ray using equation 2.18 by substituting the x and y coordinates of pixels. The similar values of t_k are searched. Then, the values of 1 are added to the corresponding elements of weight matrix that represent the positions of these pixels in the object domain.

2.3 Forward Projection of three-dimensional CT

This section presents the forward projection of three-dimensional CT; cone-beam imaging geometry. As opposed to 2D imaging, the benefit of cone-beam CT (CBCT) is volumetric visualisation of the object. An x-ray point source emanates cone-shaped radiation to irradiate an object. On the opposite side, a two-dimensional detector is located to measure the x-ray flux after being attenuated by the object [116]. An acquisition of a 2D projection is illustrated in figure 2-14. The source and detector plane rotates around the object in circular motion to collect all the projections.

CBCT have many advantages over fan-beam and parallel beam CT [93]. Firstly, it is very dose-efficient because the emitted x-rays are utilised more. Instead of 1D projection, the x-rays yield a 2D projection for each angle. Secondly, motion artefacts from patient movement are less

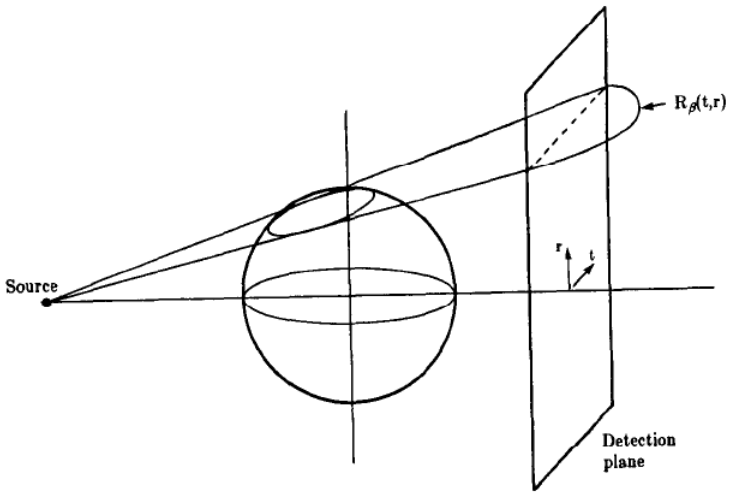


Figure 2-14: Acquisition of cone-beam CT projection [72]

affected because the data acquisition time is much shorter than that of the fan-beam CT [114], [140]. With the cone-beam CT imaging, the projections are measured through the entire object in the time it would take to measure a single slice using 2D geometries. However, the advantages come at the cost of a more complex implementation. A weight matrix for cone-beam imaging geometry requires massive space to store the values computed from the intersections of cone-beam with the voxel of an object.

For an in-depth understanding of CBCT forward projection, two matrix-based approaches are studied in this section.

Cone-beam forward projection from Tomography Image Reconstruction Software (TIRS)

The TIRS software is developed by researchers in our research group at Engineering Tomography Lab, University of Bath. The implementation of the software package is explained step-by-step in this section. The diagram of the CBCT imaging geometry is shown in the figure 2-15.

First of all, the parameters for CBCT geometry need to be defined. Some of these parameters are annotated in figure 2-15. The rest are presented as following:

- Number of acquired projections
- Number of reconstruction projections
- Scale factor
- Scan angle
- Source data path
- Start angle

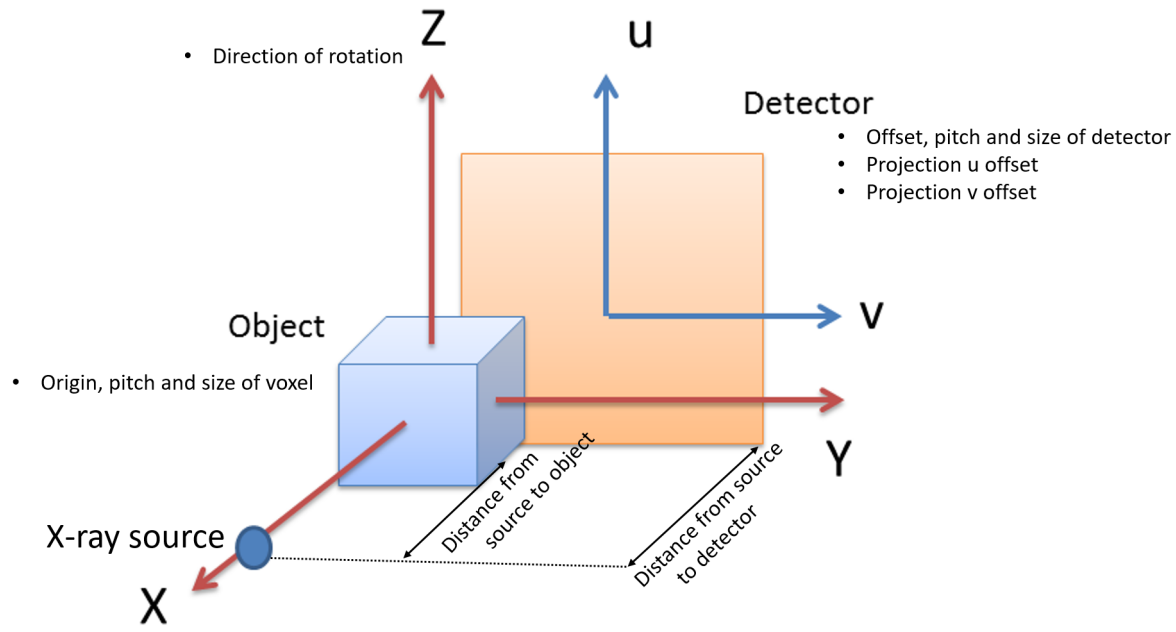


Figure 2-15: The diagram of the CBCT imaging geometry.

Next, the transform matrices from object space to projection space for all projections are computed from the cone-beam parameters.

Four types of transform include:

- Rotation transforms

$$X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(a) & -sin(a) & 0 \\ 0 & sin(a) & cos(a) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Y = \begin{pmatrix} cos(a) & 0 & sin(a) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(a) & 0 & cos(a) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Z = \begin{pmatrix} cos(a) & -sin(a) & 0 & 0 \\ sin(a) & cos(a) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Where a is an angle of rotation.

- Displacement transform

$$D = \begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Where x, y, z are coordinates in x, y, z dimensions.

- Scaling transform

$$S = \begin{pmatrix} x & 0 & 0 & 0 \\ 0 & y & 0 & 0 \\ 0 & 0 & z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Where x, y, z are coordinates in x, y, z dimensions.

- Perspective transform, perspective plane at distance f in z dimension

$$R = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

Next step, the system matrix is computed. Two inputs required for the function to compute the system matrix are transform matrix for each projection from the previous step and parameter structure. The output is the system matrix obtained for each projection. Then, these matrices are concatenated to form one system matrix A for all projections.

Firstly, the boundaries for x, y and z dimensions are defined. Empty matrices are pre-allocated to create a sparse matrix at the end. The software then loops over the size of detector in both u and v dimensions to find intersections in y and z for all x planes, intersections in x and z for all y planes and intersections in x and y for all z planes.

Coordinates of intersections are combined and sorted. The reference points are calculated by multiplying the difference between maximum and minimum coordinates with the scale factor parameter and normalise the term. Then, the row of coordinates are sorted again according to the distance from the reference points. The system matrix for each projection is created by allocating values of distance between coordinates according to coordinates' rows and columns in the object domain.

A forward projection is computed by multiplying the system matrix A with the object x, e.g. the thorax phantom as shown in figure 2-16. In this example, the size of an object is $256 \times 256 \times 256$ voxels and the detector's size is 512×512 pixels. The computed forward projection at angle 0° using the system matrix obtained from this method is shown in figure 2-17.

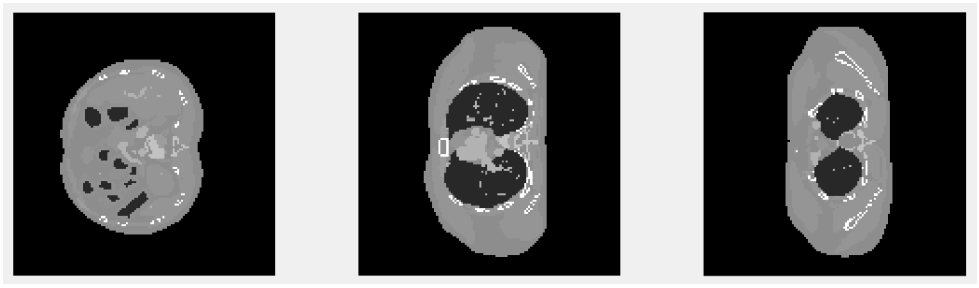


Figure 2-16: Three cross-sectional slices of the thorax phantom sized $256 \times 256 \times 256$ voxels.

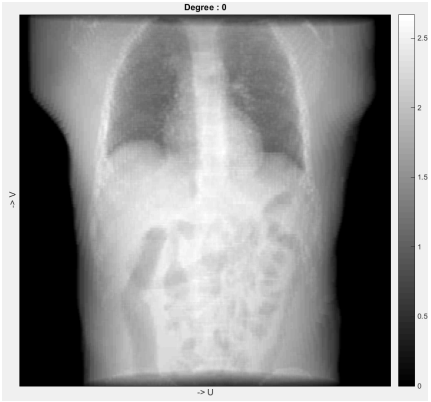


Figure 2-17: The forward projection at angle 0° of the thorax phantom sized $256 \times 256 \times 256$ voxels on the detector sized 512×512 pixels

Cone-beam forward projection based on a ray-tracing method

After several approaches to acquire forward projection in both 2D and 3D CT are studied, this method is created to ensure a solid understanding of the forward projection from CBCT, regarding a system geometry and a formation of system matrix A. The method is developed based on the ray-tracing forward projection [149], [94]. The detailed explanation of each process is presented in this section.

The parameters are first defined, in the same way as the previous method. These parameters are:

- Voxel size of object in x,y and z directions
- Real size of each object voxel in millimetres
- Real size of object in millimetres
- Pixel size of detector in u and v directions
- Real size of each detector pixel in millimetres
- Real size of detector in millimetres
- Source to detector distance in millimetres
- Source to object distance in millimetres
- Projection angle in degree
- Offset (in millimetres) of the object's centre from the origin point
- Offset (in millimetres) of the detector's centre in y and z axis (offset in x axis is specified by source to detector distance)

In a toy example of this study, the object is a cube with voxel size of $64 \times 64 \times 64$ pixels. Its real size is $256 \times 256 \times 256\ mm^3$. The detector size is 128×128 pixels with real size of $512 \times 512\ mm^2$. The CBCT system geometry being studied follows the diagram as shown in figure 2-15. The origin of X,Y,Z coordinates is at the centre of the cube object. The source-to-object distance is defined to be 1000 mm. So, the X-ray source is located at $(1000,0,0)$ on the X axis. While the source-to-detector distance is defined to be 1536 mm, the detector is then situated at $(-536,0,0)$ along the X axis, on the other side of the object.

Then, the next step is to find physical position of each detector element with reference to X,Y,Z coordinate system, i.e. finding real positions of the middle of detector pixel starting from the first pixel in the top left corner to the last pixel in the bottom right corner. An illustration how this function works is shown in figure 2-18. As mentioned earlier, the centre of detector plane is located at coordinate $(-536,0,0)$. The physical size of detector is $512 \times 512\ mm^2$ where the size of each element is $4 \times 4\ mm^2$. The position of each detector element in X,Y and Z axes of coordinates is then calculated accordingly from the first to the last element.

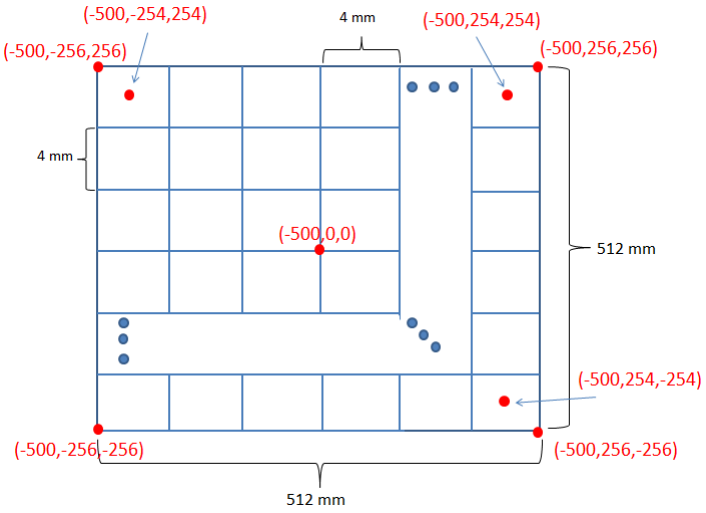


Figure 2-18: Physical positions $(x, y, z$ coordinates) of detector elements.

This process also takes into account multiple projection angles by multiplying each coordinate with a rotation matrix about Z-axis as follows.

$$R_z(\theta) = \begin{pmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

In CBCT, the ray emanates from the x-ray point source passing through the object to reach the detector plane on the other side. In order to simulate this, a ray is defined as a straight line originating from the x-ray source and ends at a detector element. This is repeated for the whole set of cone-beam geometry starting from the first ray, which ends at the first detector's element (top-left corner). The last ray is the one that ends at the last detector's element (bottom-right corner). The number of rays equals to the number of detector's elements, i.e. $128 \times 128 = 16,384$ rays, for this example.

The parametric forms of the equation of a line, as defined in equations 2.11,2.12,2.13, are used to simulate the rays. Also, the equations are used to find intersection points between each particular ray and the object by calculating any points that satisfy the equations of line.

To implement this, we start by considering the boundary of the object along the x axis. As an example in this study, the size of the object is $256 \times 256 \times 256\ mm^3$. Thus, the range of object boundary along the x axis is defined from $x = -128$ to $x = 128$ with increment of 4 (size of each voxel element), as illustrated in figure 2-19.

Recalling the 'Symmetric equations of the line' for x dimension:

$$t_x = \frac{x - x_0}{a} \tag{2.19}$$

The direction vector a is calculated by subtracting between the position of detector element and position of the x-ray source in x coordinates. Therefore, we use the equation 2.19 to compute

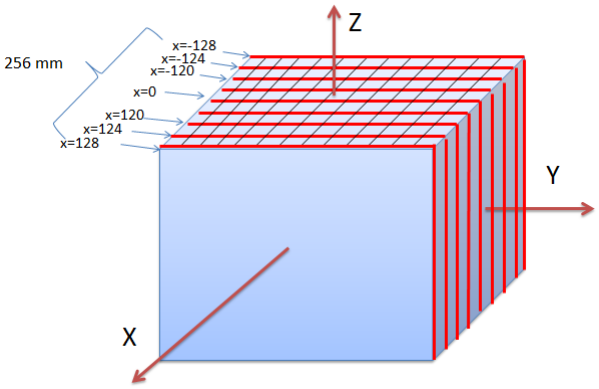


Figure 2-19: The range of object boundary along the x axis

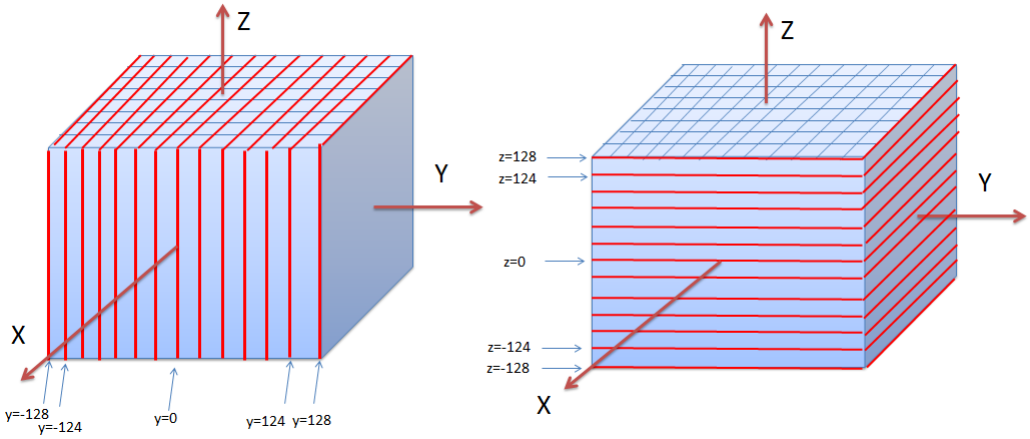


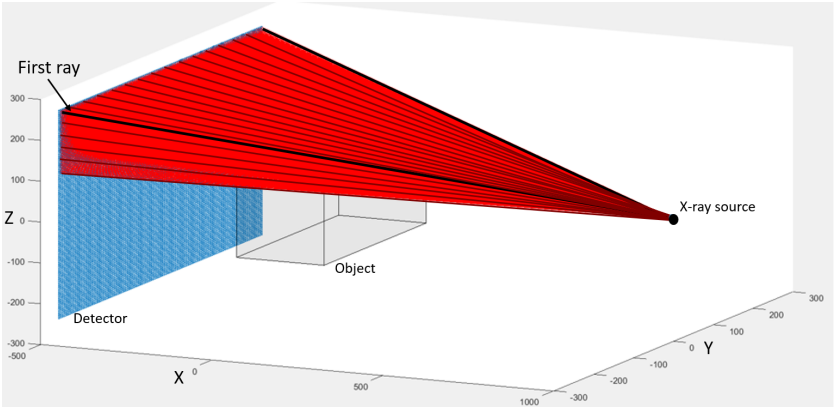
Figure 2-20: The ranges of object boundaries along y and z axes

the parameter t_x.

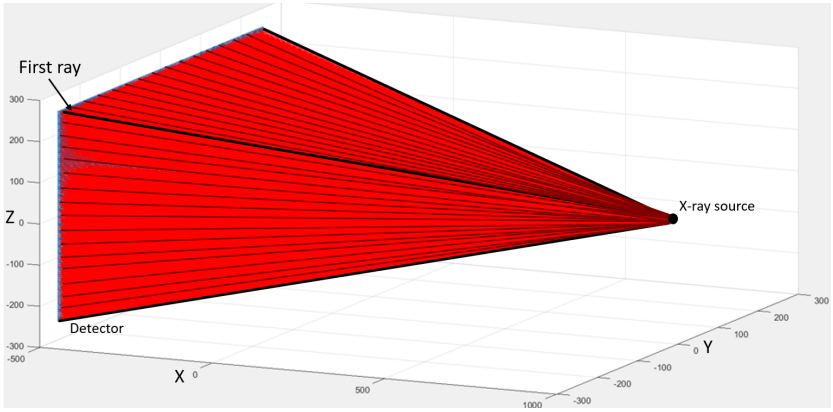
Next, we want to find the corresponding values of y and z coordinates for the intersection points from the specified values of x. These can be obtained by substituting t_x into parametric forms of equations of line for y and z dimensions, as defined in 2.12,2.13, respectively.

So far, we already have intersection points between the rays and the object boundaries along x axis. This process is repeated for the y and z axes, as illustrated in figure 2-20. In the same way as the parameter t_x, parameters t_y and t_z are computed using equation 2.19 but replacing RHS to be the variables of y and z accordingly. Then, the corresponding values of x and z for the intersection points are computed using t_y, as well as the corresponding values of x and y using t_z.

At this stage, all the intersection coordinates x,y,z between the rays and the object domain are collected. Figure 2-21 illustrates the cone-beam system in this study, showing the trajectory of each ray from the point source through the object to the detector elements. Values of parameters t_x, t_y, t_z computed from each ray that intersects with each voxel boundary in x,y and z dimensions are also collected.



(a) Intersection coordinates are sorted from the first ray starting from the point source and ending at the first detector element at the top left corner.



(b) Intersection coordinates are running through each detector element until the last one at the bottom right corner.

Figure 2-21: An illustration of the cone-beam system in this study showing the trajectory of each ray from the point source through the object to the detector elements.

Then, the values of t are sorted from minimum to maximum in each ray. The intersection coordinates are then sorted accordingly.

The system matrix A is computed based on the intersection coordinates which have been found so far. The number of rows for the matrix A equals to number of rays multiplied with number of projections. The number of columns equals to number of object's voxels which represents the index location of each element.

Figure 2-22 illustrates the index of the voxel in the example of this study, which specifies the location where one particular ray intersects with the object. The number of voxel is $64 \times 64 \times 64$ which equals to 262,144 voxels in total. The order of voxel's index starts from the first one in the upper left corner of the cube, running along every columns in the first row and then moving on to the next row and so on. When the first plane of voxels is finished, the index continues counting in the same pattern in the second plane and so on until the it reaches the last index in the lower right corner of the last plane.

Next, the length of intersection of the ray with the voxel is computed as the Euclidean dis-

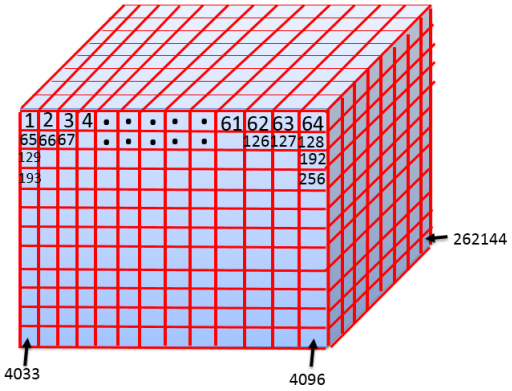


Figure 2-22: Index of the voxel starting from 1 in the upper left corner of the first plane to $64 \times 64 \times 64$ in the lower right corner of the last plane

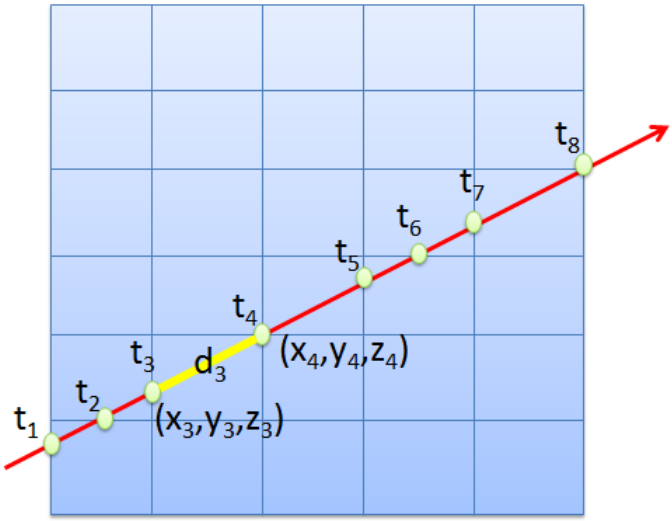


Figure 2-23: Length of intersection of one particular ray with the image voxels

tance between adjacent intersection points using the equation :

$$distance = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{2.20}$$

These intersection lengths are filled in the elements of the system matrix A, starting from the first row which stores the intersection lengths between the first ray with the object voxels. If there is no intersections, the matrix element is input the value of 0. Otherwise, the intersection lengths are stored at the corresponding voxel's location, i.e. corresponding column of the matrix. An example of one particular ray is shown in figure 2-23. This process is repeated for every ray to collect all the intersection lengths between cone-beam and the object voxels.

In the same way as the previous method, a forward projection is computed by multiplying the system matrix A computed using this method with the thorax phantom as an object x. The size of an object is $128 \times 128 \times 128$ voxels and the detector size is 256×256 pixels. Forward

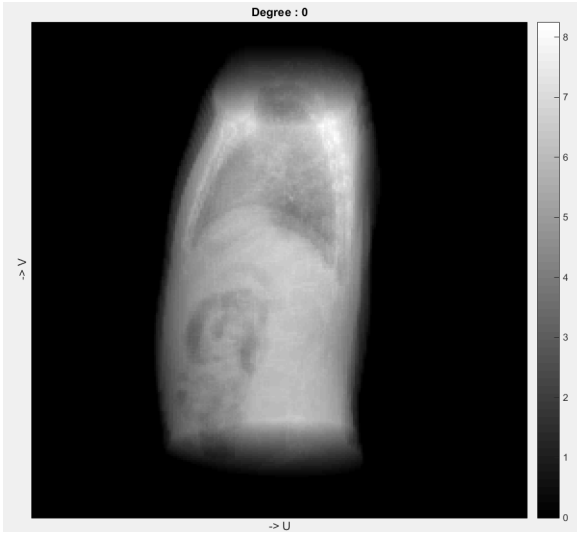


Figure 2-24: Forward projection of the thorax phantom sized $128 \times 128 \times 128$ voxels on the detector sized 256×256 pixels at projection angle 0°.

projections for angles 0° and 90° are computed and shown in figures 2-24 and 2-25, respectively.

2.4 Summary

This chapter presents an acquisition process of x-ray CT projections. The definition of line integral as well as the Central Slice Theorem are explained. The highlight of this chapter is the detailed explanation of how the forward projection is calculated, specifically how the elements of matrix A are computed based on the ray-voxel intersection. Different ways to compute forward projections for 2D CT including parallel-beam and fan-beam and 3D cone-beam imaging geometry, which is the main focus of the works in this thesis are explained.

Although the system matrix A is accurately created using the methods as explained in this chapter, the size of the matrix can be relatively massive, as one column represents each pixel of the image and one row for each ray. This is particularly true for 3D images. For example, in a typical real-world scenario, the size of 3D volume is $2048 \times 2048 \times 2048$ with 1024 projections of the detector size 2048×2048. The size of the system matrix in this example is massive, with the number of non-zero elements approximately 8×10^{12}. Thus, storing matrix A is often infeasible.

Fortunately, the matrix A is rarely used as a single matrix in the iterative algorithms. Instead, the matrix-vector multiplication forms are computed, namely Ax and $A^T b$. The projection (Ax) represents an integral of image over the straight x-ray paths to the detector. The backprojection ($A^T b$) represents the smearing of the detector data over the image following the straight paths back to the direction of the source. With the nature of these 2 matrix-vector multiplication forms, it is possible to compute and accelerate them using graphic processing unit (GPU) computing since the parallelism is the important key feature of GPU. The GPU accelerated projection and backprojection make it possible to implement and develop iterative algorithms with

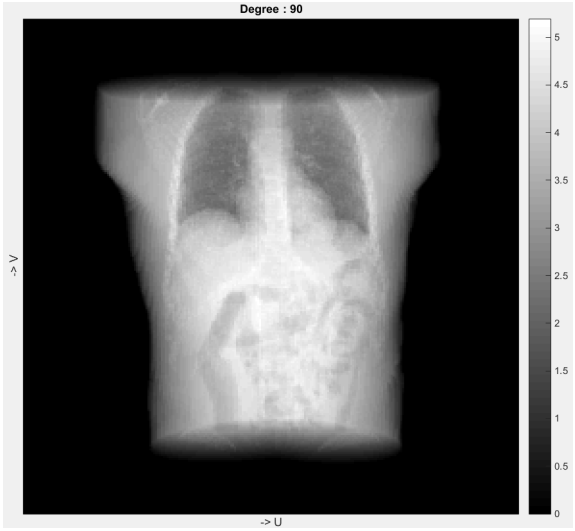


Figure 2-25: Forward projection of the thorax phantom sized $128 \times 128 \times 128$ voxels on the detector sized 256×256 pixels at projection angle 90°.

high computational efficiency within a practical time frame.

There are several advanced toolboxes available that include a number of iterative reconstruction algorithms and GPU-accelerated forward and back-projection operators such as ASTRA [127], RTK [109] and 3D CB CT MATLAB [74]. Among these toolboxes, ASTRA and RTK are the ones that are most complete. However, for the purpose of developing new algorithms, these toolboxes are less suitable to work with because of their infrastructure in low-level programming language. Instead, the Tomographic Iterative GPU-based Reconstruction (TIGRE) toolbox proposed by Biguri *et al* [9] is a better alternative for developing new algorithms. It is a MATLAB-CUDA toolbox which offers a modular, geometrically flexible and easy to use toolbox for cone and parallel beam CT with GPU acceleration. Numerous CT reconstruction algorithms are available in the TIGRE toolbox, as well as GPU accelerated Ax and $A^T b$.

The comparisons of some existing iterative algorithms in this thesis employed fundamental algorithms available in the TIGRE toolbox. The development of new algorithms, later on in this thesis, is based on GPU accelerated projection and backprojection available in this toolbox.

Chapter 3

X-ray CT image reconstruction algorithms

After the x-ray CT projections from multiple views are obtained, another important part of CT reconstruction is the backprojection process to retrieve an image. The backprojection process distributes the projection data back along the same path that the projection data were formed. The equal intensity of projection data in one angle is smeared back evenly along the path, on to the imaging grid. This process is repeated for all angles. After this process, the result of backprojection is not quite similar to the original image. Instead, it is an original image with blurring effect, especially around the centre of the imaging grid. This results from the excess intensities adding up when the projection data are smeared back in all angles.

The filtering process is then implemented to get rid of the blurring effect of backprojection. The projection data is filtered first before being backprojected along the same path. As a result, the blurring effect is reduced and the result image is more clear. This reconstruction algorithm is called Filtered Backprojection (FBP) algorithm, which is explained in more detail in the next section.

This chapter explains a variety of image reconstruction algorithm, which is a mathematical process that dissolves the blurring effect and produces a non-overlapped image of the original image.

Generally, there are two mains categories of image reconstruction algorithms; analytical and iterative reconstruction algorithms. The implementations, as well as advantages and disadvantages of these algorithms are discussed in this chapter.

3.1 Filtered backprojection (FBP)

The most common approach used in CT reconstruction is FBP algorithm, which is based on the Central Slice Theorem. Referring to figure 2-4 in the previous chapter, the central region of the Fourier space represents low frequencies components. It can be seen from the figure that there is a high density building up around the origin of the Fourier space, which causes the blurring effect as mentioned earlier. The mathematical filters can be used to eliminate this

effect. The Fourier transform $P(\omega,\theta)$ of the projection data is multiplied by $|\omega|$, which is ramp filter transfer function before implementing inverse Fourier transform. After the projection data is filtered, they are backprojected to obtain the exact original image $f(x,y)$.

FBP algorithm can be implemented by following the steps below:

(1) Compute the 1D Fourier transform of the projection data $p(s,\theta)$ to obtain $P(\omega,\theta)$

(2a) Multiply the Fourier transform $P(\omega,\theta)$ with a ramp filter $|\omega|$ to obtain $Q(\omega,\theta)$

$$Q(\omega,\theta) = P(\omega,\theta) \times H(\omega)$$

Then, compute the 1D inverse Fourier transform to obtain $q(s,\theta)$

(2b) Equivalently, instead of multiplication in the ω domain, the convolution in s domain can be implemented to obtain the ramp-filtered data $q(s,\theta)$

$$q(s,\theta) = p(s,\theta) * h(s)$$

(3) Backproject the filtered data to obtain reconstructed image $f(x,y)$

The FBP algorithm is described as

$$f(x,y) = \int_0^{\pi} \int_{-\infty}^{\infty} P(\omega,\theta)|\omega| e^{j2\pi\omega(xcos\theta+ysin\theta)} d\omega d\theta \tag{3.1}$$

Let $Q(\omega,\theta) = |\omega|P(\omega,\theta)$

$$f(x,y) = \int_0^{\pi} \int_{-\infty}^{\infty} Q(\omega,\theta) e^{j2\pi\omega(xcos\theta+ysin\theta)} d\omega d\theta \tag{3.2}$$

Applying 1D inverse Fourier transform to obtain

$$f(x,y) = \int_0^{\pi} q(xcos\theta + ysin\theta, \theta) d\theta \tag{3.3}$$

An approximation of the 3D FBP algorithm is called the FDK algorithm, named after Feldkamp, Davis and Kress from their work proposed in [44]. The FDK algorithm is the predominant algorithm used in most of the commercial CT reconstruction softwares for both medical and industrial applications. This is due to its small computational power required for the implementation. In the scenario where it is possible to take a complete set of projections from all around the object, the accurate reconstruction can be obtained from the FDK algorithm.

However, in cases where the available projection data are limited due to many constraints such as scaning geometry, radiation exposure or imaging hardware, the FDK algorithm performs less efficient and often produces result image with severe artefacts [62].

When the projection data is sparse, the reconstruction problem is extremely ill-posed, or sensitive to measurement noise [31], [82]. Theoretically, an accurate reconstruction from sparse projection data is possible [96]. However, a priori information on the image is required to obtain a good quality reconstruction.

3.2 Iterative algebraic reconstruction algorithms

As an alternative to FDK algorithm, iterative algebraic reconstruction algorithms are used. The reconstruction algorithms in this category are capable of handling undersampled data and generate reconstructed images with higher quality.

Recall the system of linear equations as appeared in equation 1.1, and recast again in the following equation:

$$\sum_{n=1}^{N} a_{mn} x_n = b_m + e \tag{3.4}$$

where $x \in \mathbb{R}^N$ is a vector representing an N dimensional image voxel in lexicographical order ($n = 1, 2, ..., N$), $b \in \mathbb{R}^M$ is the measured data from M measurements ($m = 1, 2, ..., M$), e is the additive noise associated with the measurement and a_{mn} is the weighting factor representing the contribution that the n^{th} image voxel has to the m^{th} line integral. The majority of a_{mn} values will be zero as only a small number of image voxel contribute to a give ray-sum.

An iterative algorithm can be considered as a closed-loop process. In each iteration, the algorithm starts with an initialised/estimated image. Then, a projection is computed from the image and compared with the measured projection data. The discrepancy between the estimated and measured projection data is used to update the estimated image for the next iteration. The process is repeated iteratively until the image with the desired properties is achieved. In addition, with the iterative algorithms, it is possible to incorporate prior information about the image into consideration. Some traditional iterative algorithms are discussed in the following sections.

3.2.1 Algebraic Reconstruction Technique (ART)

One of the most frequently used methods among the iterative type algorithms is the method proposed by Kaczmarz [70], also known as the Algebraic Reconstruction Technique (ART) [52]. This algorithm is a projection-based algorithm aiming to solve for the solution that represents the closest approximation to the object function from the available projections [92]. For the ART algorithm, the image is updated following this equation:

$$x_n^{(k)} = x_n^{(k-1)} + \lambda \frac{b_m - \sum_{n=1}^{N} x_n^{k-1} a_{mn}}{\sum_{n=1}^{N} a_{mn}^2} a_{mn} \tag{3.5}$$

where λ is a relaxation parameter, m is the iteration index. The term $\sum_{n=1}^{N} x_n^{k-1} a_{mn}$ represents the computed ray-sum for the m^{th} ray based on the $(k-1)^{th}$ solution for the estimated image.

Equivalently, equation 3.5 can be re-written as:

$$x_n^{(k)} = x_n^{(k-1)} + \lambda \Delta x_n^{(k-1)} \tag{3.6}$$

where

$$\Delta x_n^{(k-1)} = \frac{b_m - \sum_{n=1}^{N} x_n^{(k-1)} a_{mn}}{\sum_{n=1}^{N} a_{mn}^2} a_{mn} \tag{3.7}$$

represents the correction term which is computed by calculating the difference between the measured projection data and the computed projection from the estimated image. The difference is normalised by $\sum_{i=1}^{N} a_{mn}^2$. Then, this value is multiplied with the weight a_{mn}. The correction term is then assigned to all the image cells in the m^{th} ray. For the ART algorithm, the reconstruction grid needs to be sequentially updated on a ray-by-ray basis. The ART algorithm updates the n^{th} imaging voxel using the m^{th} equation.

The pseudo code of the ART algorithm is shown in Algorithm 1

Algorithm 1: Algebraic Reconstruction Technique (ART)

```
Initialisation ;
x_0 : an initial image in R^N;
λ : relaxation parameter;
End of initialisation step;
while convergence not reached do
    x_n^{k,0} = x_0;
    for m=1:M do
        Δx_n^{k,m} = (b_m − Σ_{n=1}^{N} x_n^{k,m−1} a_mn) / (Σ_{n=1}^{N} a_mn^2) a_mn;
        x_n^{k,m+1} = x_n^{k,m} + λ Δx_n^{k,m};
    end for
    x_n^{k+1} = x_n^{k,M};
    Until a stopping criterion is met
end while
```

Generally, in practical application, the size of the system matrix A is massive. Thus, updating an image on a ray-to-ray basis is not practical and is a main disadvantage of the ART algorithm. In addition, it was observed by Anderson and Kak in [4] that there is a striping effect occurs when using ART method due to the updating of the reconstruction grid for every ray. This leads to the implementation of the next algorithm, the Simultaneous Iterative Reconstruction Technique (SIRT).

3.2.2 Simultaneous Iterative Reconstruction Technique (SIRT)

The SIRT algorithm is proposed to address the erroneous reconstruction as found from using the ART algorithm [48]. The SIRT algorithm updates the reconstruction grid using all the data in the measurement simultaneously. It goes through all the equations and updates the n^{th} imaging voxel at the end of each iteration using the average value of all the computed changes for that imaging voxel.

For SIRT algorithm, the image is updated following this equation:

$$x_n^{(k)} = x_n^{(k-1)} + \lambda \frac{1}{M_n} \sum_{m=1}^{M} \frac{b_m - \sum_{n=1}^{N} x_n^{(k-1)} a_{mn}}{\sum_{n=1}^{N} a_{mn}^2} a_{mn} \tag{3.8}$$

Equivalently, the equation 3.8 can be expressed using the correct term in the equation 3.7 as following:

$$x_n^{(k)} = x_n^{(k-1)} + \lambda \frac{1}{M_n} \sum_{m=1}^{M} \Delta x_n^{(k-1)} \tag{3.9}$$

where M_n is number of non-zero elements in the column vector a_n. The correction terms are accumulated for each voxel and the reconstruction grid is only updated after each iteration. Each iteration is then just a single update of the image. The pseudo code of the SIRT algorithm is shown in Algorithm 2

Algorithm 2: Simultaneous Iterative Reconstruction Technique (SIRT)

```
Initialisation ;
x_0 : an initial image in R^N;
λ : relaxation parameter;
M_n : number of non-zero elements in the column vector a_n;
End of initialisation step;
while convergence not reached do
    x_n^{k,0} = x_0;
    Δx_n^{k,0} = 0;
    for m=1:M do
        Δx_n^{k,m} = (b_m − Σ_{n=1}^N x_n^{k,m−1} a_mn) / (Σ_{n=1}^N a_mn^2) a_mn;
        Δx_n^{k,m+1} = Δx_n^{k,m} + Δx_n^{k,m−1} ;
    end for
    x_n^{k+1} = x_n^{k,M} + λ (1/M_n) Δx_n^{k,M};
    Until a stopping criterion is met
end while
```

The SIRT algorithm can suppress the noise artefacts, which results in better reconstruction. However, the problem with the algorithm is the long time it needs for convergence. This leads to a development of Simultaneous Algebraic Reconstruction Techniques (SART), which is a middle ground between ART and SIRT algorithms.

3.2.3 Simultaneous Algebraic Reconstruction Technique (SART)

The SART algorithm is developed by Anderson and Kak [4]. The correction terms are computed using simultaneously all data from each x-ray projection, in the same way as the SIRT algorithm. However, the reconstruction grid is updated for each single voxel using an accumulated correc-

tion term that is due to all pixel in the projection. The image is updated following this equation:

$$x_n^{(k)} = x_n^{(k-1)} + \lambda \frac{\sum_{m=1}^{M_p} \left(\frac{b_m - \sum_{n=1}^{N} x_n^{(k-1)} a_{mn}}{\sum_{n=1}^{N} a_{mn}^2} \right) a_{mn}}{\sum_{m=1}^{M_p} a_{mn}^2} \tag{3.10}$$

which can also be expressed as:

$$x_n^{(k)} = x_n^{(k-1)} + \lambda \frac{\sum_{m=1}^{M_p} \Delta x_n^{(k-1)}}{\sum_{m=1}^{M_p} a_{mn}^2} \tag{3.11}$$

where M_p is total number of rays in one particular angle, $M = Mp \times p$ and p is number of projections.

The pseudo code of the SART algorithm is shown in Algorithm 3

Algorithm 3: Simultaneous Algebraic Reconstruction Technique (SART)

```
Initialisation ;
x_0 : an initial image in R^N;
λ : relaxation parameter;
End of initialisation step;
while convergence not reached do
    x_n^{k,0} = x_0;
    for p = 1 : M_p do
        Δx_n^{k,p} = (b_p − Σ_{n=1}^{N} x_n^{k,p−1} a_pn) / (Σ_{n=1}^{N} a_pn^2) a_pn;
    end for
    x_n^{k+1} = x_n^{k,M} + λ (Σ_{M_p=1}^{M} Δx_n^{k,M_p}) / (Σ_{M_p=1}^{M} a_{M_p n});
    Until a stopping criterion is met
end while
```

The SART algorithm can be expressed in another format as:

$$x^{(k)} = x^{(k-1)} + \lambda V^{-1} A^T W (b - A x^{(k-1)}) \tag{3.12}$$

- V is diagonal with n^{th} diagonal element : $V_{nn} = \sum_{m=1}^{M} |A_{mn}|$
- W is diagonal with m^{th} diagonal element $W_{mm} = \frac{1}{\sum_{n=1}^{N} |A_{mn}|}$

Another variant of SART is Ordered Subsets-SART (OS-SART) [19], [130], where the update of the reconstruction grid is done using block-based methods. The projection data is divided into groups or subsets. The update is performed for each subset, instead of the complete set of available projections. The idea of implementing ordered subsets (OS) to reduce the reconstruction times was originally proposed for emission tomography in [75], [60]. Later on, the idea was transferred to transmission methods in [85]. The general form of the OS-SART algorithm is:

$$x^{(k)} = x^{(k-1)} + \lambda V^{-1} [A(\beta)]^T [W(\beta)] ([b(\beta)] - [A(\beta)] x^{(k-1)}) \tag{3.13}$$

- $W(\beta)$ is diagonal with m^{th} diagonal element: $[W(\beta)]_{mm} = \frac{1}{\sum_{n=1}^{N} |A_{mn}(\beta)|}$
- b is divided into B blocks: $b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_B \end{bmatrix} \in \mathbb{R}^M, b_\beta \in \mathbb{R}^{M_\beta}, \beta \in [1, B]$ and $M = \sum_{\beta=1}^{B} M_\beta$
- A is divided into B blocks: $A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_B \end{bmatrix} \in \mathbb{R}^{M \times N}, A_\beta \in \mathbb{R}^{M_\beta \times N}$

The pseudo code of the OS-SART algorithm is shown in Algorithm 4

Algorithm 4: Ordered Subsets Simultaneous Algebraic Reconstruction Technique (OS-SART)

```
Initialisation ;
x_0 : an initial image in R^N;
λ : relaxation parameter;
End of initialisation step;
while convergence not reached do
    x_n^{k,0} = x_0;
    for β = 1 : B do
        Δx_n^{k,β} = (b_β − Σ_{n=1}^{N} x_n^{k,β−1} a_{βn}) / (Σ_{n=1}^{N} a_{βn}^2) a_{βn};
    end for
    x_n^{k+1} = x_n^{k,M} + λ (Σ_{M_B=1}^{M} Δx_n^{k,M_B}) / (Σ_{M_B=1}^{M} a_{M_B n});
    Until a stopping criterion is met
end while
```

In summary, all ART-based methods perform better than the analytical approach with sparse data and irregular sampling of projection acquisition [7]. This is because the analytical approach such as FBP algorithm assumes a continuum of acquisition positions, while the ART-based methods model the geometry of the acquisition process better.

3.2.4 Conjugate Gradient Least Squares (CGLS)

The CGLS is a well-known and standard tool for finding the least squares solution, x^*, for the sparse systems of equations with a symmetric, positive definite coefficient matrix [10], [55].

$$x^* = min||Ax - b||_2 \qquad subject\ \ to\ \ x \in \kappa_k(A^TA, A^Tb) \tag{3.14}$$

where $\kappa_k(A^TA, A^Tb)$ is the Krylov subspace associated with the normal equations:

$$A^TAx = A^Tb \tag{3.15}$$

The CGLS algorithm performs i steps of the conjugate gradient algorithm, which is applied implicitly to the normal equations. In each step, the algorithm minimises the next biggest eigenvector of the residual of the normal equations' 2-norm. As a result, the CGLS algorithm converges very fast when comparing to other iterative methods [77]. The pseudo code of the CGLS algorithm is presented in Algorithm 5.

Algorithm 5: Conjugate Gradient Least Squares (CGLS)

Initialisation ;
x_0 : an initial image;
$r^0 = b - Ax^0$;
$p^0 = A^T r^0$;
$\psi = \frac{||A^T r^0||^2}{Ap^0||^2}$;
End of initialisation step;
while *convergence not reached* **do**
    for *k=1:i* **do**
        $r^{(k)} = r^{(k-1)} - \psi^{(k-1)} A p^{(k-1)}$;
        $\beta^{(k-1)} = \frac{||A^T r^{(k)}||^2}{A^T r^{(k-1)}||^2}$;
        $p^{(k)} = A^T r^{(k)} + \beta^{(k-1)} p^{(k-1)}$;
        $\psi^{(k)} = \frac{||A^T r^{(k)}||^2}{Ap^{(k)}||^2}$;
        $x^{(k)} = x^{(k-1)} + \psi^{(k-1)} p^{(k-1)}$;
    end for
    Until a stopping criterion is met
end while

However, there is some loss of accuracy due to the implicit use of the cross-product $A^T A$. In addition, the CG type methods such as LSQR [100] require storing the whole weighting matrix A before the actual calculation, which make the methods of this type rarely used in the CBCT reconstruction because the usual size of the matrix A for CBCT is extremely large.

3.3 Convergence study between SIRT, SART, OS-SART and CGLS algorithms

In this section, the convergence and computational time of the 4 iterative algorithms explained earlier are compared. The test was performed using the synthetic dataset from the XCAT phantom [115]. The voxel size of the data is 128^3 and the detector size is 256^2. The number of projection angles of the data set is 100 projection views, uniformly distributed over 360°. The testing computer is an Intel Core i7-4930K CPU@3.40GHz with 32 GB RAM and GPU: NVDIA GeForce GT 610. These algorithms being tested are derived from the TIGRE toolbox: a MATLAB-GPU toolbox for CBCT image reconstruction [9]. The maximum number of iteration for each algorithm is set at 1,000 iterations.

Figure 3-1 shows the normalised residual of the reconstructed images from 4 algorithms for each iteration until the algorithms finish their implementation. The normalised residual is

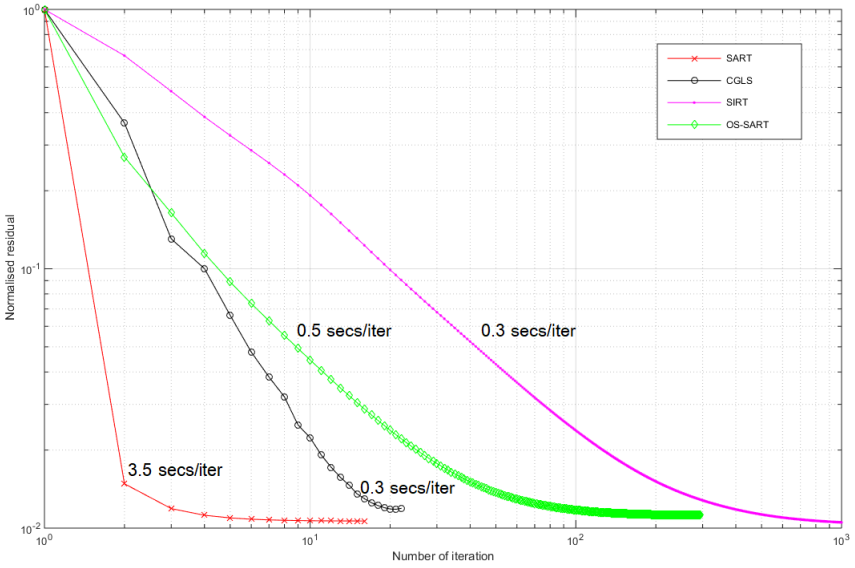


Figure 3-1: The normalised residual of the reconstructed images from SART, CGLS, SIRT and OS-SART algorithms. The normalised residual was computed and presented per iteration until the algorithms stopped their implementation. The SART, CGLS and OS-SART algorithms stopped due to divergence at iteration number 16,21 and 291, respectively. The SIRT algorithm stopped at the maximum number of iteration.

computed following the equation.

$$\frac{||Ax-b||_2}{||b||_2}$$

According to figure 3-1, the SART, CGLS and OS-SART algorithms stopped due to divergence at iteration number 16,21 and 291, respectively. The SART algorithm stopped when the maximum number of iteration is reached. Comparing the computational time taken for one iteration of each algorithm, SART algorithm takes the longest time per iteration, i.e. $\approx$ 3.5 seconds. OS-SART algorithm takes $\approx$ 0.5 seconds/iteration. SIRT and CGLS algorithms take $\approx$ 0.3 seconds/iteration. It is worth noting that the computational time presented in this convergence study is based on the setting and power of the computer using in the experiments as mentioned earlier. Different versions of TIGRE toolbox can also have some effects on the computational time.

Although the normalised residual of SART algorithm is the lowest among others, the computational time per iteration is the longest. The SIRT algorithm takes the shortest time per iteration, but it is very slow to converge to the same level of the residual as those of the SART and OS-SART algorithms. As a middle ground, the OS-SART algorithm converges to approximately the same normalised residual level, with $\approx$6 times faster in term of the computational time comparing to the SART algorithm.

To analyse the results of the test visually, the cross-sectional slices of the reconstructed images from 4 algorithms are shown in figure 3-2. In addition, the plot of one dimensional

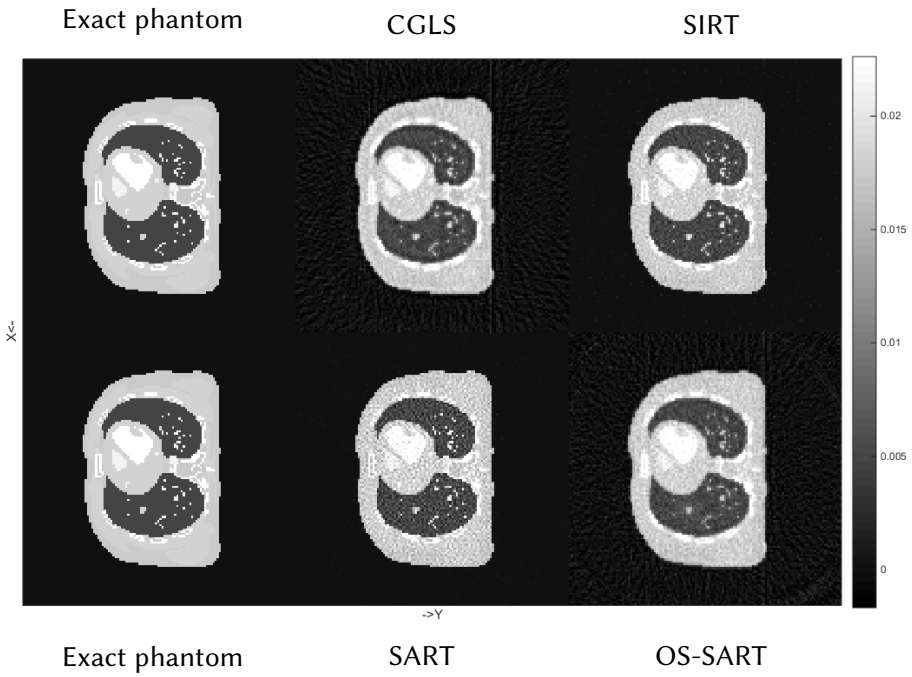


Figure 3-2: Cross-sectional slices of the reconstructed images from CGLS, SIRT, SART and OS-SART algorithms.

profiles along an arbitrary row of the reconstructed images is shown in figure 3-3.

According to figures 3-2 and 3-3, there is no single algorithm with an outstanding result among the 4 algorithms, when comparing to the exact phantom image. However, the result from the OS-SART algorithm in the one-dimensional profiles plot seems to be better aligned with the plot of the exact phantom image. This implies that the reconstruction image obtained from the OS-SART is more similar to the exact phantom image.

More experiments were studied to better understand the behaviour of each algorithm. In the previous experiment, the 4 algorithms were implemented until they satisfy the stopping criteria. In other words, the algorithm iterates until the algorithm starts to diverge, i.e. the normalised residual of the estimated image in the current iteration is greater than that of the previous iteration, in order to stop the algorithm at the point of minimum residual error. The other stopping criterion is when the algorithm reaches the pre-defined maximum number of iteration.

In the next experiments, the first stopping criterion of the 4 algorithms is suppressed and the algorithms are forced to implement until they reach the maximum number of iterations (1,000 iterations). The objective is to observe how the normalised residuals of the reconstructed images change per each iteration of the algorithms. The normalised residual plots of the 4 algorithms per each iteration until iteration number 1,000 is displayed in figure 3-4.

It can be seen from figure 3-4 that, when the algorithms are forced to iterate until iteration number 1,000, some algorithms start to diverge and end up having the reconstructed image with the higher residual error than the minimum point. This is obvious for the CGLS algorithm. After iteration number 16 of the CGLS algorithm, level of normalised residual per iteration consider-

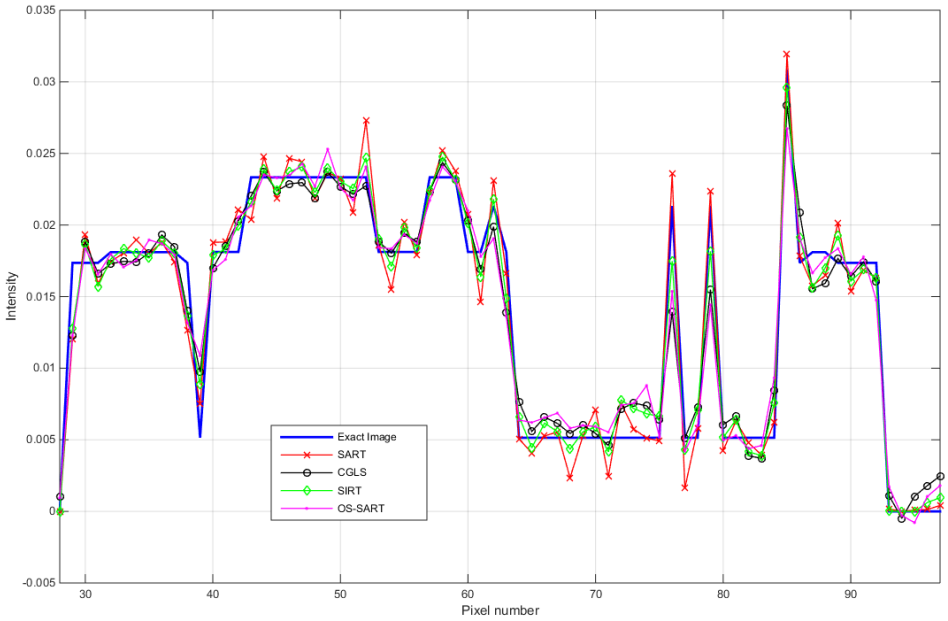


Figure 3-3: One-dimensional profiles plot of the 4 iterative reconstruction algorithms, in comparison with the exact image.

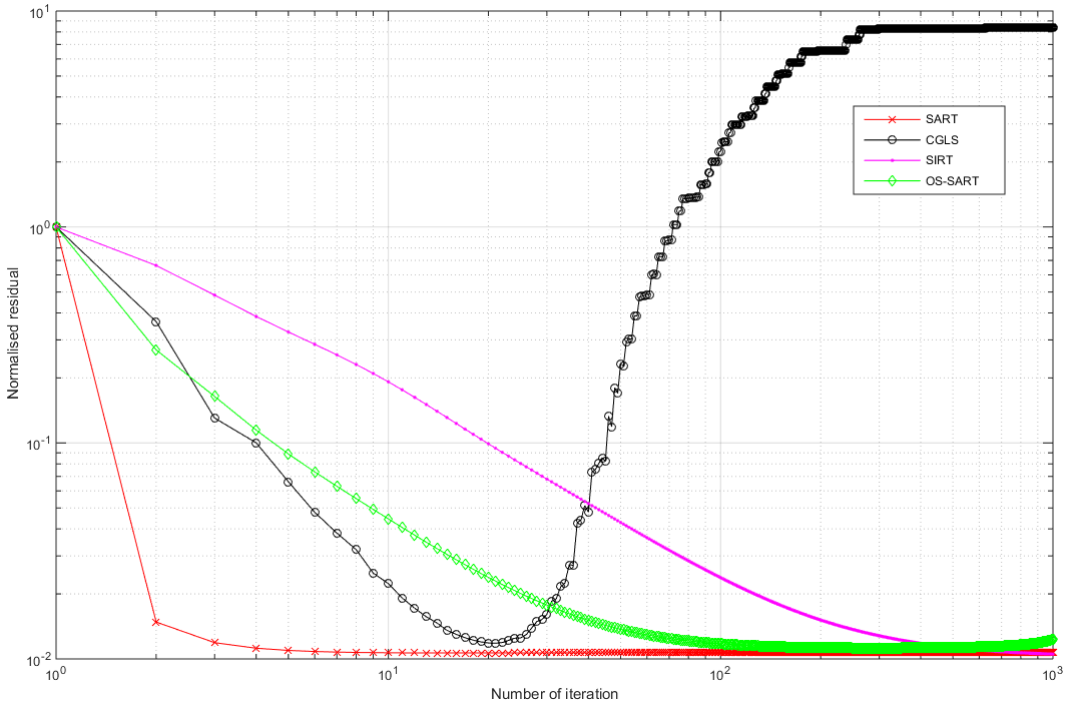


Figure 3-4: The normalised residual of the reconstructed images from SART, CGLS, SIRT and OS-SART algorithms per each iteration. All the algorithms are forced to stop at the maximum number of iterations of 1,000.

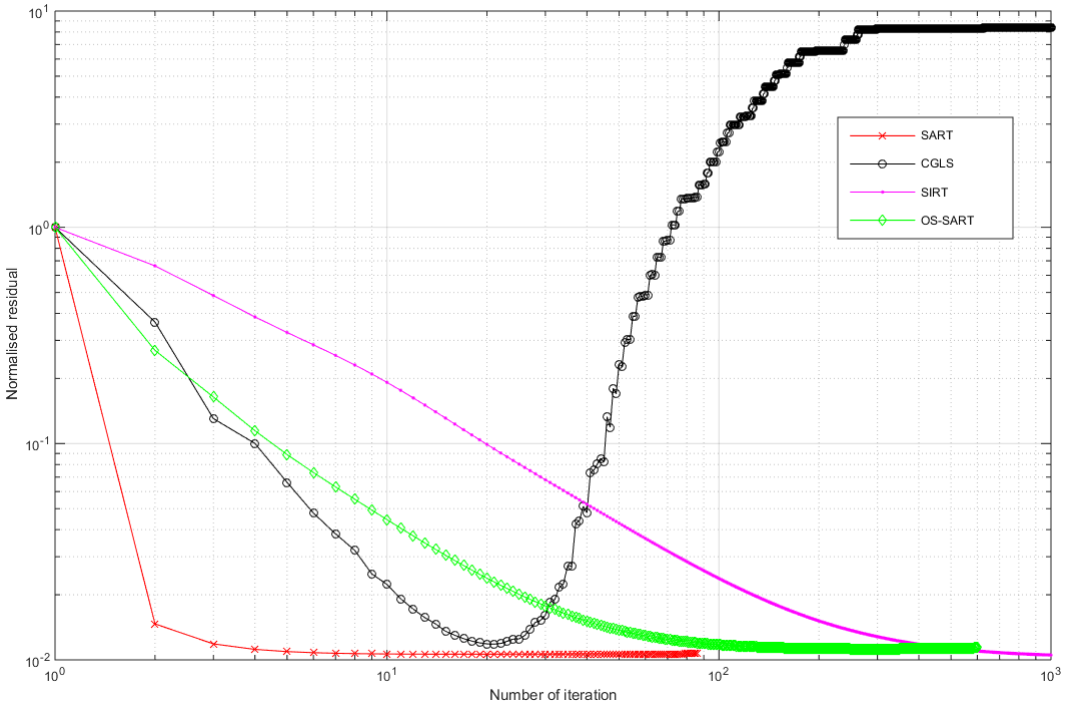


Figure 3-5: The normalised residual of the reconstructed images from SART, CGLS, SIRT and OS-SART algorithms per each iteration. All the algorithms are compared within the same running time of 5 minutes. The CGLS and SIRT algorithms are implemented for 1,000 iterations. The OS-SART and SART algorithms are implemted for 600 and 85 iterations, respectively.

ably increased. The final reconstructed image of the CGLS after it reaches the maximum number of iteration contains much higher normalised residual error, compared to the one of iteration number 16. Although it is not clearly visible, the same phenomenon occurred for the SART and OS-SART algorithms. These two algorithms still iterated until the maximum number of iteration was reached. The final reconstructed images contain slightly higher normalised residual errors than those of the minimum point of error.

Next, the performance of the 4 algorithms is compared based on the running time. Instead of specifying the same maximum number of iteration (1,000 iterations as shown in figure 3-4), the 4 algorithms were specified the same running time of 5 minutes, as a benchmark. The computational time per each iteration among all algorithms is different, as presented earlier and annotated in figure 3-1. This leads to different number of iterations within 5 minutes time frame of each algorithm. Approximately 1,000 iterations of the CGLS and SIRT algorithms can be implemented within 5 minutes since they require ≈0.3 seconds per iteration. Six hundreds iterations of OS-SART algorithm can be implemented (≈0.5 seconds/iteration). Lastly, 85 iterations of SART algorithm can be implemented (≈3.5 seconds/iteration). Again, the stopping criteria was suppressed in this experiment to let the algorithms iterate to the specified iteration number. The normalised residuals of the reconstructed images per each iteration upto 5 minutes running time of the 4 algorithms are shown in figure 3-5.

The results from the same 5-minute running time presented in figure 3-5 are rather similar to

those of the same maximum number of iteration presented in 3-4. The differences between these two experiments are regarding to what were used as benchmarks to compare, either maximum number of iteration or running time. As shown in figure 3-5, the 3 algorithms namely CGLS, SART and OS-SART were iterated beyond the minimum normalised residual points and stopped after 5 minutes with different number of iterations. The normalised residual errors of SIRT algorithm still decreased until the algorithm stopped at the same time limit.

From these experiments of the convergence study in this section, we can see that the stopping criteria of the iterative algorithms play an important role to the performance of the algorithms. They stop the algorithms at the point of minimum residual, to avoid the divergence of the algorithms. The performance of the iterative algorithms can be compared on the basis of running time. However, the iterative algorithms implement iteration-by-iteration and obtain the reconstructed image of the current iteration at the end of each iteration. The time of running each algorithm will then be based on how many iterations can be implemented within the specified time-frame.

It is also important to discuss here that there are many factors that could affect the computational time of each iterative algorithm such as computer power, GPU power, structure of the MATLAB codes, version of TIGRE toolbox used in this work, etc. Thus, comparison based on iterations can represent better perspective of the iterative algorithms' performance.

The final concluding of this chapter is that minimising the residual errors of the reconstructed image is not the only factor to achieve a good reconstruction result. The image reconstruction task can be further improved by formulating it as an optimisation problem with a regularisation. More details will be presented in the next chapter.

3.4 Summary

This chapter presents the detailed implementation of x-ray CT image reconstruction algorithms, both analytical approach which is the Filtered backprojection (FBP) algorithm and iterative algebraic approaches such as Algebraic Reconstruction Technique (ART), Simultaneous Iterative Reconstruction Technique (SIRT), Simultaneous ART (SART) and Conjugate Gradient Least Squares (CGLS) algorithms.

The FBP algorithm is the most commonly used and well-known reconstruction algorithm across the commercialised softwares. This is because the implementation of FBP algorithm is straightforward and less computationally expensive compared to the iterative algorithms. However, the FBP algorithm requires a lot of projection data to reconstruct good quality of image. In the limited data scenario, the iterative algorithms show superior results to the analytical approach.

The convergence and performance of the 4 iterative reconstruction algorithms are studied in this chapter. Among the 4 algorithms, the reconstructed image from the OS-SART algorithm has slightly better quality compared to other algorithms, when observing the visual inspection and one-dimensional profile plots.

However, the reconstruction results can be improved further by using regularisation tech-

niques, especially the total variation (TV) minimisation, which is the main focus of the algorithm proposed in this thesis. More details of the regularisation algorithms will be discussed in the next chapter.

Chapter 4

Total variation regularisation algorithms

The amount of radiation dose delivered to patients from X-ray CT is the great concern in medical imaging. With limited angular configurations, the projection angles are sampled more sparsely in an attempt to reduce the radiation or due to physical limitations. The reconstruction from insufficient data degrades the reconstructed image as a consequence. This makes the inverse problem of CT reconstruction often ill-posed. In mathematical aspect, the number of data samples is not sufficient to uniquely determine the values of image vector by using direct inversion method.

In order to reconstruct an image from available measurements, a numerous number of iterative algorithms have been extensively used and developed during the past decades. The differences among the algorithms are the constraints used for the image function, the cost function being minimised and the scheme of implementation. Some well-known examples of iterative algorithms have been shown in the previous chapter.

4.1 Mathematical formulation

Let us state again an image reconstruction problem, which can be formulated as solving a system of simultaneous linear equations:

$$\sum_{n=1}^{N} A_{mn} x_n = b_m + e, \quad m \in [1, M] \tag{4.1}$$

where $x \in \mathbb{R}^N$ is a vector representing an N dimensional image voxel in lexicographical order of size N,$b \in \mathbb{R}^M$ is the measured data from M measurements and e is the additive noise associated with the measurement. A is the system matrix.

The problem of estimating image x from measured data b is called a linear inverse problem. In most scenarios of practical interest, specifically for the CT reconstruction problem, the linear inverse problem is often ill-posed, i.e. the system matrix A is very ill-conditioned. Therefore, it requires some kind of regularisation or prior information about the target image.

Formulating the linear inverse problem as a constrained optimisation problem is one way

of the regularisation. The image reconstruction problem from X-ray CT measurement can be formulated as a constrained optimisation problem in the form:

$$\min_x \phi(x) \qquad subject\ \ to\ \ \|Ax-b\|_2 \leq \varepsilon \tag{4.2}$$

Notice that the constrained optimisation formulation include explicit constraint on the variables that the unknowns must satisfy. ϕ is the regulariser or regularisation function and $\varepsilon \geq 0$ is an error bound that defines the amount of acceptable error between predicted and observed projection data. In real practice, it is impossible to always obtain the reconstructed image that is perfectly consistent with the data due to several factors such as modelling errors, noise and x-ray scattering [122]. Therefore, the constraint in equation 4.2 only require that the reconstructed image yields projection data that are within a given l_2 distance ε of the actual projection data. The constrained optimisation problem in the form of equation 4.2 minimises the regularisation function ϕ with subject to the constraint that the error between the predicted and observed projection data lies within the pre-defined ε value.

Another category of the optimisation problem is an unconstrained optimisation problem, which arises as a reformulation of the constrained optimisation problem. In the unconstrained optimisation problem, the constraints are replaced by penalisation terms in the objective function that have the effects of discouraging constraint violations [99]. The unconstrained optimisation problem can be written in the form:

$$\min_x \frac{1}{2}\|Ax-b\|_2^2 + \tau\phi(x) \tag{4.3}$$

where τ is the regularisation parameter and ϕ is the regulariser or regularisation function. The first term of the unconstrained problem in equation 4.3 is the data fidelity term, which aims to minimise the error between the predicted and observed projection data with an additional regularisation term. The formulation of unconstrained optimisation problem makes sense when there is a unique image that minimises the data fidelity term. By adding the regularising penalty term, the image sacrifices data fidelity for image regularity. The balance between two terms is controlled by the regularisation parameter τ.

When there is a number of images with equivalent data fidelity, the regularisation term takes on the role of selecting the image with desired property out of the set of feasible images with the same data fidelity. The formulation of constrained optimisation problem in equation 4.2 makes this process more naturally separated. The resulting image is ensured to satisfy the constraint, while also achieving the desired property. As can be seen in the equation 4.3, the data fidelity constraint is formulated as an inequality constraint to reflect multiple sources of data inconsistency such as noise, x-ray scatter and a simplified data model. These factors make it impossible to always find the image that is absolutely consistent with the data. In addition, some common physical constraints/assumptions such as image positivity can be easily incorporated with the formulation of constrained optimisation problem.

The idea of accurate reconstruction using a limited number of measurements came from the discovery of compressed sensing (CS) approach, which was first proposed by [13], [35].

The CS approach has a strong impact on CT reconstruction after the successful performance as shown in the studies [24], [120]. Among the possible candidates for CS penalty, the non-smooth regularisers such as the total variation (TV) and $l_1(\phi(x) = ||x||_1)$ norms are popular and powerful choices for CT imaging.

The prior knowledge about the structure of CT images can be incorporated to push the reconstruction algorithms towards a specific solution among multiple possibilities. For example, the medical tomographic images are constant over flat regions which can be assumed as regions within an organ. The flat regions are separated by sharp edges that may only occur at boundaries of internal structures.

The gradient image in 3D is defined as:

$$|\nabla(x)| = \sqrt{(x_{ijk} - x_{ij-1k})^2 + (x_{ijk} - x_{i-1jk})^2 + (x_{ijk} - x_{ijk-1})^2} \tag{4.4}$$

where i, j, k are indices of image voxel in three dimensions.

The $l1$ norm of the gradient image is called the total variation (TV) of the image. The idea of using TV penalty methods was introduced by Rudin, Osher and Fatemi (ROF) in 1992 [110]. It was observed that minimising the TV, with subject to other conditions regarding variance and mean of the image, better preserves the edges information of the image. It has been used in image processing for image denoising while preserving the edges [129], [21], [8], [29].

The TV norm in 3D case is derived as the following equation.

$$||x||_{TV} = \sum_{ijk} \sqrt{(x_{ijk} - x_{ij-1k})^2 + (x_{ijk} - x_{i-1jk})^2 + (x_{ijk} - x_{ijk-1})^2} \tag{4.5}$$

The iterative algorithms are developed by taking advantage of this sparseness of the gradient image. The TV norm of the image is used as a regularisation function in the optimisation formulation. In 2004, the algorithm for minimising the TV of an image is proposed by Chambolle in [20]. The TV has been commonly used as a regularising criterion for solving inverse problems in a number of studies [21], [50], [88], [86].

The constrained optimisation problem as presented in equation 4.2 can be formulated again as in the following equation, by minimising the image TV norm as a regularisation function:

$$x^* = \arg\min_x ||x||_{TV} \tag{4.6}$$

with subject to the following two constraints:

(A) data fidelity constraint

$$||Ax - b||_2 \leq \varepsilon \tag{4.7}$$

where ε is an error bound that defines the amount of acceptable error between predicted and observed projection data.

(B) non-negativity constraint

$$x \geq 0 \tag{4.8}$$

The second constraint is applied to the image voxels based on the assumption that negative

attenuation values are not possible.

4.2 Karush Kuhn-Tucker (KKT) conditions

The necessary conditions for a constrained minimisation to find an optimal image is called the Karush Kuhn-Tucker (KKT) conditions. These conditions are a generalisation of the Lagrange multiplies for inequality constraints. The description of the KKT conditions explained in this section is from [122]. See for examples in [99] and [11].

Firstly, the Lagrangian is defined for the current problems:

$$L = \|\vec{x}\|_{TV} + \lambda_0(\|A\vec{x} - \vec{b}\|^2 - \varepsilon^2) - \vec{\lambda} \cdot \vec{x} \tag{4.9}$$

The Lagrangian in equation 4.9 combines the objective function with each constraint multiplied by a multiplier λ_i. The data constraint is a single constraint, thus it receives a single-value multiplier λ_0 in the Lagrangian. For the non-negativity constraint, it is a constraint on each image voxel. Therefore, there is a multiplier λ_i for each image voxel, where i represents the index of each image voxel. The set of multipliers for the non-negativity constraints is $\vec{\lambda}$, which is a vector of the same size as the image.

The following two inequality constraints are imposed on the Lagrange multipliers.

(1) Non-negativity:

$$\lambda_i \geq 0 \tag{4.10}$$

(2) Complementarity:

$$h_i(\vec{x})\lambda_i = 0 \tag{4.11}$$

Where $i = 0, 1, ..., N_{voxels}$ and h_i represents alternative form of the inequality constraints $h_i(\vec{x}) \leq 0$:

$$\begin{aligned} h_0(\vec{x}) &= \|A\vec{x} - \vec{b}\|^2 - \varepsilon^2 \leq 0 \\ h_i(\vec{x}) &= -x_i \leq 0; \qquad\qquad i \in [1, N_{voxels}] \end{aligned} \tag{4.12}$$

Only when an inequality constraint is violated, its corresponding h_i becomes positive. The complementarity condition allows the Lagrange multiplier λ_i to be non-zero only when its corresponding inequality constraint is active, i.e. when equation 4.12 is satisfied with equality.

Therefore, the conditions for a potential solution $\vec{x}$ are equations 4.10,4.11,4.12 and $\nabla_{\vec{x}} L = 0$. The differential operator is defined as:

$$\nabla_{\vec{x}} Q(\vec{x}) = \sum_i \partial_{x_i} Q(\vec{x}) \vec{\delta}_i \tag{4.13}$$

where $\vec{\delta}_i$ is the Knonecker delta, which is a zero image everywhere, but ones at the ith voxel.

The gradient of the Lagrangian can be written as:

$$\begin{aligned}\nabla_{\vec{x}} L &= \nabla_{\vec{x}} \|\vec{x}\|_{TV} + \lambda_0 \nabla_{\vec{x}} h_0(\vec{x}) + \sum_{i=1}^{N_{voxels}} \lambda_i \nabla_{\vec{x}} h_i(\vec{x}) \\ &= \nabla_{\vec{x}} \|\vec{x}\|_{TV} + 2\lambda_0 A^T (A\vec{x} - \vec{b}) - \vec{\lambda} \\ &= 0\end{aligned} \tag{4.14}$$

Since the non-negativity constraints are only active in zero-value voxels, the Lagrange multipliers are zero for strictly positive voxels, due to complementarity. With this knowledge, the following indicator function:

$$\vec{x}_{indicator} = \begin{cases} 1, & \text{if } \vec{x} \neq 0 \\ 0, & \vec{x} = 0 \end{cases} \tag{4.15}$$

can be used to simplify the gradient of the Lagrangian equation in 4.14 to:

$$\nabla_{\vec{x}} L = diag(\vec{x}_{indicator})(\nabla_{\vec{x}} \|\vec{x}\|_{TV} + \lambda_0 \nabla_{\vec{x}} h_0(\vec{x})) = 0 \tag{4.16}$$

This equation can be separated into 2 vectors, namely the TV and data constraint gradients:

$$\begin{aligned}\vec{d}_{TV} &= diag(\vec{x}_{indicator})(\nabla_{\vec{x}} \|\vec{x}\|_{TV}) \\ \vec{d}_{data} &= diag(\vec{x}_{indicator})(\nabla_{\vec{x}} h_0(\vec{x}))\end{aligned} \tag{4.17}$$

These two vectors play an important role to assess the optimality of an image estimate $\vec{x}$. Hence, the KKT conditions state that an image $\vec{x}$ is optimal when the two vectors $\vec{d}_{TV}$ and $\vec{d}_{data}$ point in exactly opposite direction. In practice, the image is checked if it satisfies the inequality constraints and the two vectors are pulling in the opposite direction by computing the dot product. The cosine of the angle between $\vec{d}_{TV}$ and $\vec{d}_{data}$ is computed:

$$cos\theta = \frac{\vec{d}_{TV} \cdot \vec{d}_{data}}{|\vec{d}_{TV}||\vec{d}_{data}|} \tag{4.18}$$

The cosine of angle is used as a test of optimality. If the two vectors are exactly opposite to each other, the angle between them is 180° and the $cos\theta$ is -1.0. However, this value is rather difficult to achieve in practice and requires a large number of iterations. It is reported in [122] that ,empirically, there is only little significant changes in the image when the $cos\theta$ is below -0.5. Thus, the aim is to attain $cos\theta$ in the neighbourhood of -0.99.

The algorithm to solve the constrained optimisation problem checks the direction of the vectors $\vec{d}_{TV}$ and $\vec{d}_{data}$ to ensure that even if the data constraint is satisfied, only the optimal image regarding both TV norm and data fidelity is chosen as a solution.

4.3 Alternating direction method of multipliers (ADMM)

From this section onwards, the methods to solve the optimisation problem of the CT reconstruction are reviewed. The first one is the Alternating direction method of multipliers (ADMM), which was first introduced in the mid 1970s by Gabay and Mercier [47], Glowinski and Marrocco [49] with roots in the mid 1950s.

The ADMM method decomposes a large-scale optimisation problem into a sequence of small local sub-problems. The method utilises the benefits of dual decomposition and augmented Lagrangian methods for constrained optimisation, where the objective function is separable.Solutions to small sub-problems are coordinated to find a solution to large-scale global problem.

A general form of an ADMM problem is (with f, g convex):

$$\begin{aligned} &minimise && f(x)+g(z) \\ &subject\ \ to && Ax+Bz=c \end{aligned}$$

Two set of variables with separable objective.

$$L_\rho(x,z,y) = f(x)+g(z)+y^T(Ax+Bz-c)+(\frac{\rho}{2})||Ax+Bz-c||_2^2$$

• ADMM

$$\begin{aligned} x^{k+1} &:= argmin_x L_\rho(x,z^k,y^k) && //x-minimisation \\ z^{k+1} &:= argmin_z L_\rho(x^{k+1},z,y^k) && //z-minimisation \\ y^{k+1} &:= y^k+\rho(Ax^{k+1}+Bz^{k+1}-c) && //dual\ \ update \end{aligned}$$

In this section, the way the ADMM method can be applied to solve the CT reconstruction problem of the following form is presented.

$$\min_x \sum_{i=1}^{n^2} \|D_i x\|_2 + \frac{\mu}{2}\|Ax-b\|_2^2 \tag{4.19}$$

Where x is an image $x \in \mathbb{R}^{n^2}$, $D_i x \in \mathbb{R}^2$ represents the 1st order finite difference of image x at pixel i, $\|D_i x\|_2$ is the variation of image x at pixel i and $TV(x) = \sum_{i=1}^{n^2} \|D_i x\|_2$.

The description in this section followed the materials explained in [124]. In the problem under determine, both the system matrix A and the finite difference operators have structures, an algorithm is then designed to take advantage of problem structures. The formulation in 4.19 is transformed by the authors of [132] to the following form, for this purpose.

$$\min_{x,y} \left\{ \sum_i \|\mathbf{y}_i\| + \frac{\mu}{2}\|Ax-b\|^2 : \mathbf{y}_i = D_i x, i = 1,...,n^2 \right\} \tag{4.20}$$

Where $\mathbf{y}_i \in \mathbb{R}^2$ is an auxiliary vector. The objective function in equation 4.20 is separable and

the constraints are linear. For convenience, let $y = (y_1; y_2) \in \mathbb{R}^{2n^2}$, where y_1 and y_2 are vectors of length n^2 satisfying $((y_1)_i; (y_2)_i) = \mathbf{y}_i \in \mathbb{R}^2$ for $i = 1, ..., n^2$.

The augmented Lagrangian function of equation 4.20 is defined as:

$$L_A(x, y, \lambda) \triangleq \sum_i \left(\|\mathbf{y}_i\| - \lambda^T (\mathbf{y}_i - D_i x) + \frac{\beta}{2} \|\mathbf{y}_i - D_i x\|^2 \right) + \frac{\mu}{2} \|Ax - b\|^2 \tag{4.21}$$

Start with $x = x^k$ and $\lambda = \lambda^k$, the alternating minimisation applied to equation 4.20 yields the iterative scheme as following:

$$\begin{cases} y^{k+1} \leftarrow argmin_y L_A(x^k, y, \lambda^k) \\ x^{k+1} \leftarrow argmin_x L_A(x, y^{k+1}, \lambda^k) \\ \lambda^{k+1} \leftarrow \lambda^k - \beta(y^{k+1} - Dx^{k+1}) \end{cases} \tag{4.22}$$

The minimisation of $L_A(x^k, y, \lambda^k)$ with respect to y is equivalent to n^2 two-dimensional problem of the following form:

$$\min_{\mathbf{y}_i \in \mathbb{R}^2} \|\mathbf{y_i}\| + \frac{\beta}{2} \left\| \mathbf{y}_i - \left(D_i x^k + \frac{1}{\beta} (\lambda^k)_i \right) \right\|^2, \quad i = 1, 2, ..., n^2 \tag{4.23}$$

The solution of the problem in 4.23 can be given by the following two-dimensional shrinkage [132] [142]:

$$\mathbf{y}_i^{k+1} = max\left\{ \left\| D_i x^k + \frac{1}{\beta} (\lambda^k)_i \right\| - \frac{1}{\beta}, 0 \right\} \frac{D_i x^k + \frac{1}{\beta} (\lambda^k)_i}{\|D_i x^k + \frac{1}{\beta} (\lambda^k)_i\|}, \quad i = 1, 2, ..., n^2 \tag{4.24}$$

Where it is assumed that $0 \cdot (0/0) = 0$.

Fixing $\lambda = \lambda^k$ and $y = y^{k+1}$ (recall that y is a re-ordering of $\mathbf{y}_i, i = 1, 2, ..., n^2$), the minimisation of L_A with respect to x is a least squares problem and the corresponding normal equations can be written as following:

$$\left(D^T D + \frac{\mu}{\beta} A^T A \right) x = D^T \left(y^{k+1} - \frac{1}{\beta} \lambda^k \right) + \frac{\mu}{\beta} A^T b \tag{4.25}$$

Where $D \triangleq \left(D^{(1)}; D^{(2)} \right) \in \mathbb{R}^{2n^2 \times n^2}$ is the global first-order finite difference operator. The equation 4.25 can be solved in 3 steps:

- First, the right-hand side vector in equation 4.25 is computed and a forward fast Fourier transform (FFT) is applied.
- Second, $\mathscr{F} x^{k+1}$ is obtained by componentwisely dividing the eigenvalues of $D^T D + \frac{\mu}{\beta} A^T A$, where $\mathscr{F}$ denotes 2D discrete Fourier transform.
- Third, the 2D inverse discrete Fourier transform is applied to $\mathscr{F} x^{k+1}$ to obtain the new iterate x^{k+1}

The λ is updated by the following equation:

$$\lambda^{k+1} = \lambda^k - \beta(y^{k+1} - Dx^{k+1}) \tag{4.26}$$

At this point, the ADMM approach for image reconstruction to solve the problem of the form 4.19 or equivalently 4.20 can be presented in the Algorithm 6.

Algorithm 6: Alternating direction method of multipliers (ADMM)

```
Initialisation ;
Input b, A, μ > 0, β > 0 and λ^0;
Set x = b and λ = λ^0;
End of initialisation step;
while not converged do
    1) Compute y^{k+1} according to equation 4.24 for given (x^k, λ = λ^k) ;
    2) Compute x^{k+1} by solving equation 4.25 ;
    3) Update λ^{k+1} by equation 4.26 ;
    Until stopping criteria are met
end while
```

The ADMM algorithm terminates by relative change in image x according to the following equation:

$$\frac{\|x^{k+1} - x^k\|}{max\{\|x^k\|, 1} \} < \varepsilon \tag{4.27}$$

Where $\varepsilon > 0$ is an accepted error.

4.4 Projected subgradient method (PSM)

Another method to solve the constrained minimisation problem is the Projected Subgradient Method (PSM). The root of this method in the unconstrained case was proposed in 1985 by Shor [118], while the root in the constrained case was proposed around 1969 by Polyak [104], [103]. The implementation of PSM for constrained minimisation has been extensively investigated [111], [97]. This method repeatedly interlaces subgradient steps for the objective function with projections onto the intersection of closed and convex constraints sets, to regain feasibility.

The description of the PSM in this section followed the materials explained in [18]. First of all, the formation of constrained minimisation problems being considered is as following:

$$minimise\Big\{\phi(x)|x \in C\Big\} \tag{4.28}$$

Where $\phi : \mathbb{R}^J \to \mathbb{R}$ is an objective function and $C \subseteq \Omega$ is a given feasible set. The Ω is a non-empty subset of the J-dimensional Euclidean space $\mathbb{R}^J$.

The optimisation problem to solve the X-ray CT image reconstruction problem can be for-

mulated as following:

$$minimise\Big\{\phi(x)|Ax=b \ \ and \ \ 0\leq x\leq 1\Big\} \tag{4.29}$$

As suggested in Table 4.1 of [58], the box constraints $0\leq x\leq 1$ are natural for the image reconstruction problem of X-ray CT and reasonable for human body's tissues. Thus, Ω for this particular problem can be defined as $\Omega=\Big\{x\in\mathbb{R}^J|0\leq x\leq 1\Big\}$. The choice of C in equation 4.28 is of the type specified as the following equation, where $\{C_l\}_{l=1}^{L}$ is a family of constraints and C_l is non-empty closed convex subset of $\mathbb{R}^J$.

$$C=\bigcap_{l=1}^{L}C_l \tag{4.30}$$

with $L=I+1, C_i=\{x\in\mathbb{R}^J|\langle a^i,x\rangle=b_i\}$ for $i=1,2,...,I$ and $C_{I+1}=\Omega$. The choice of the objective function ϕ is TV norm in equation 4.5.

The PSM generates a sequence of iterations, i.e. $\{x^k\}_{k=0}^{\infty}$ using the following recursion formula:

$$x^{k+1}=P_C(x^k-t_k\phi'(x^k)) \tag{4.31}$$

Where $t_k>0$ is a step size and $\phi'(x^k)\in\partial\phi(x^k)$ is a subgradient of ϕ at x^k. P_C denotes the orthogonal (least Euclidean norm) projection onto the set C. In this problem, a non-summable diminishing step-length is adopted. The form is $t_k=\gamma_k/\|\phi'(x^k)\|$, where $\gamma\geq 0$, $\lim_{k\to\infty}\gamma_k=0$ and $\sum_{k=0}^{\infty}\gamma_k=\infty$.

The pseudocode of the PSM algorithm is shown in Algorithm 7.

It is reported in [18] that a major disadvantage of the PSM is that the method needs to perform an orthogonal projection onto the feasible set of the constrained minimisation problem in each iterative step. In case that the feasible set is not simple to project onto, the projection requires an independent inner-loop calculation to minimise the distance from a point to the feasible set. This process is costly and affects overall effectiveness of the PSM.

even be in C, and, even if it is in C, it is most unlikely to be the minimizer of ϕ over C. The second problem in real-world applications comes from the fact that the constraints, derived from the real-world problem, may not be consistent (e.g., because they come from noisy measurements), and so C is empty.

4.5 Adaptive-steepest-descent projection onto convex sets (ASD-POCS)

The pioneering work in this field is the algorithm proposed by Sidky *et al* [121], which is a combination of gradient descent on the TV objective function as in the equation 4.6 and projection onto convex sets (POCS) to enforce the data and the positivity constraints as in the equations 4.7 and 4.8. The aim is to find the solution to the constrained, optimisation problem. However,

Algorithm 7: Projected Subgradient Method (PSM)

Initialisation ;
Select a point $x^0 \in \mathbb{R}^J$;
Select integers K and M and use two real number variables **curr** and **prev**;
Set **curr**=$\phi(x^0)$ and **prev**=**curr** ;
Set $x^k$the current iterate End of initialisation step;
while *Stopping criteria not met* **do**
    Calculate a subgradient of ϕ at x^k, i.e. $\phi'(x^k) \in \partial\phi(x^k)$;
    $t_k = \frac{k^{-1/4}}{\|\phi'(x^k)\|_2}$; // Calculate a step-size
    $q^k = x^k - t_k\phi'(x^k)$;
    $x^{k+1} = \arg\min_x \left\{ \frac{1}{2}\|x - q^k\|^2 | Ax = b \quad and \quad 0 \le x \le 1 \right\}$; // Calculate the next iterate as the projection of q^k onto C
    if $\phi(x^{k+1} \le$ **curr**$)$ **then**
        curr $= \phi(x^{k+1})$;
    end if
    if $k \ \ mod \ \ K = 0$ **then**
        if **prev** $-$ **curr** $<$ **prev**$/M$ **then**
            Stop ; // Check stopping criteria
        end if
        else
            prev $=$ **curr** ;
        end if
    end if
    Until stopping criteria are met
end while

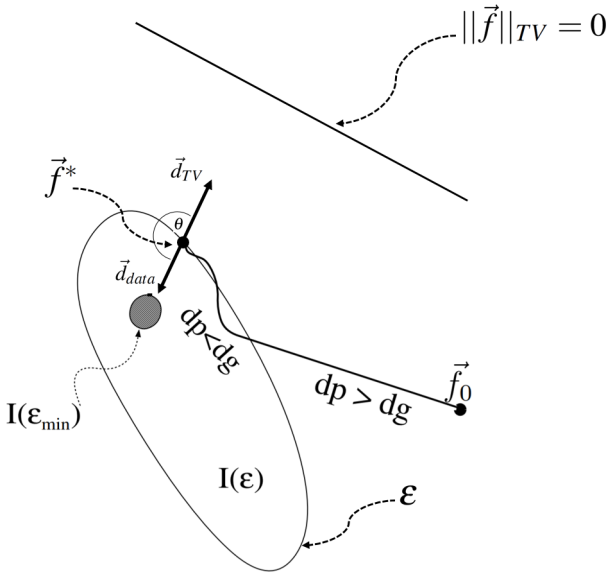


Figure 4-1: The illustration of the ASD-POCS algorithm showing the image trajectories during the iteration of the algorithm. This figure is adapted from [122].

the algorithm proposed in [121] did not explicitly consider data inconsistency and only designed for the case of ε = 0 for the equation 4.7. Their improvement on the algorithm was proposed later on in [122] with the algorithm called the Adaptive-steepest-descent projection onto convex sets (ASD-POCS) algorithm. The algorithm aims to find the solutions for the equations 4.6, 4.7 and 4.8 for a given value of ε. As the name implies, the strength of the TV minimisation update in the ASD-POCS algorithm is adaptively controlled according to the data constraint update. The pseudocode of the ASD-POCS algorithm is shown in Algorithm 8. The ASD-POCS algorithm consists of 2 steps. The Simultaneous Algebraic Reconstruction Technique (SART) is used to improve data consistency. The first step is POCS where the SART and basic projection enforces positivity are implemented. The second step is the minimisation of TV norm using adaptive-steepest-descent method.

The illustration of the ASD-POCS algorithm, which explains the image trajectories during the iteration of the algorithm is shown in the figure 4-1. The ultimate goal of the ASD-POCS algorithm is to provide the image with minimum TV that satisfies a given error tolerance. The oval represented by $I(\varepsilon)$ is all images that lie within a specified error tolerance. The shaded circle represented by $I(\varepsilon_{min})$ are the images whose projection have the minimum distance compare to the given projection data. Firstly, the algorithm starts with an initial image $\vec{f}_0$, which is represented by the small dot outside the oval. The thin curve stemming from this dot shows the image trajectory for the ASD-POCS algorithm. Starting from the initial image or when the currently estimated image is outside of the $I(\varepsilon)$, the magnitude of change due to POCS (dp) is adaptively controlled to be greater than the magnitude of change due to TV-steepest descent (dg). Once the currently estimated image is inside the $I(\varepsilon)$, dg is adaptively controlled to be greater than dp. This is to ensure that the image which has already satisfied an error tolerance

Algorithm 8: Adaptive-Steepest-Descent Projection Onto Convex Sets (ASD-POCS) algorithm

Initialisation ;
Set ε,β,β_{red},ng,α,α_{red},r_{max};
$\vec{x} = 0$;
End of initialisation step;
while *Stopping criteria not met* **do**
  $\vec{x}_{prev}$ = $\vec{x}$;
  for n_{angles} **do**
    $\vec{x} = \vec{x} + \beta V^{-1} A^T W(\vec{b} - A\vec{x})$; // SART update
  end for
  $\beta = \beta \times \beta_{red}$;
  $\vec{x}$=max$(0,\vec{x})$; // Enforce positivity constraint
  $\vec{x}_{out} = \vec{x}$;
  $\varepsilon_{current} = ||A\vec{x} - \vec{b}||$;
  $dp = ||\vec{x} - \vec{x}_{prev}||$; // Change in data constraints
  if *First iteration* **then**
    $dtv = \alpha \times dp$; // Initialisation of TV hyperparameter
  end if
  $\vec{x}_{prev}$ = $\vec{x}$;

  for *number of sub-iteration=1:ng* **do**
    $\vec{dx} = \nabla_{\vec{x}} ||\vec{x}||_{TV}$;
    $\hat{dx} = \frac{\vec{dx}}{||\vec{dx}||}$;
    $\vec{x} = \vec{x} - dtv \cdot \hat{dx}$; // TV minimisation update
  end for
  $dg = ||\vec{x} - \vec{x}_{prev}||$; // Change in TV update
  if $(dg > r_{max} \times dp)$ ***and*** $\varepsilon_{current} > \varepsilon$ **then**
    $dtv = dtv \times \alpha_{red}$
  end if

  $cos\theta = \frac{\vec{dp} \cdot \vec{dg}}{||\vec{dp}|| \cdot ||\vec{dg}||}$;
  if $(cos\theta < -0.9$ ***and*** $\varepsilon_{current} < \varepsilon)$ ***or*** $\beta < 0.0005$ **then**
    Stop ; // Check stopping criteria
  end if
  Until stopping criteria are met
end while

will have a minimum TV. The optimal solution, i.e. image within $I(\varepsilon)$ with the minimum TV, will generally lies on the boundary of the $I(\varepsilon)$ oval. This is because it is the point that the solution is within the user specified ε. Apart from that, the optimal solution is found at the point where both TV and data gradients ($\vec{d}_{TV}$ and $\vec{d}_{data}$) point in opposing direction. This is the proof of the optimality, as explained earlier in details in the section of KKT conditions (section 4.2). It is to be observed that the optimal image estimate $\vec{f}^*$ is located on the boundary of the $I(\varepsilon)$ oval with the minimum distance from the line representing zero TV norm of the image ($||\vec{f}||_{TV}$). The ASD-POCS algorithm yields an image that is close to the minimum TV image within projection-data tolerance ε. The POCS and TV-steepest descent steps are implemented alternately. The step-sizes for these 2 steps are controlled adaptively. The thin curve in figure 4-1 represents the complete image trajectory of the algorithm.

4.6 Adaptive-weighted Total Variation (AwTV) norm as an objective function

Despite the success of the ASD-POCS algorithm, a number of drawbacks arise which lead to the development of many algorithms later on, based on the principles proposed in the ASD-POCS algorithm. The first and the foremost problem is the over-smoothing in the reconstructed image, which is frequently seen due to the assumption of isotropic edge property in calculating the TV term [80]. The low-contrast structures tend to be smoothed out by the TV regularisation, which poses a great challenge for the TV methods. The edges are significant structural information of the image. Hence, the edge preservation is a critical requirement in many clinical analysis especially in Image-Guided Radiation Therapy (IGRT).

One way to address this problem is to use priors other than conventional TV norm, as expressed in 4.5, to improve preservation of fine details. There have been a number of studies to overcome this limitation. One of those is the TV-based edge preserving (EPTV) model, which is proposed by Tian *et al* [125]. The objective of the EPTV regularisation model is to to perform smoothing only on the non-edge region of the image, in order to better preserve the edges. This is realised by introducing a penalty weight vector ω in the definition of the TV term, as expressed in the following equations:

$$||x||_{EPTV} = \int \omega(f(x))||\nabla f(x)||_1 dx, \tag{4.32}$$

$$\omega(f(x)) = exp\left[-\left(\frac{||\nabla f(x)||_1}{\delta})^2\right]\right. \tag{4.33}$$

Equation 4.32 is convex. Where δ controls an amount of smoothing being applied to the image voxels at edges, relative to non-edge voxels. The edges of image is determined by computing the gradient image using equation 4.4.

Using the EPTV model, Tian *et al* also proposed the EPTV algorithm in [125], which is implemented as presented in Algorithm 9.

Algorithm 9: TV-based edge preserving (EPTV) model algorithm

Initialisation ;
x^0 : an initial image;
M : maximum number of iteration;
δ : amount of smoothing being applied to the image voxels at edges;
End of initialisation step;
while *Stopping criteria not met* **do**
  for *k=1:M* **do**
    Solve: $v^{(k+1)} = argmin_x ||Ax^{(k)} - b||_2^2$; // CGLS solver
    Solve: $x^{(k+1)} = argmin_x \frac{\pi}{2}||x^{(k)} - v^{(k+1)}||_2^2 + ||x||_{EPTV}$; // Gradient Descent
    Update: $\omega(x^{(k+1)}) = exp\left[-\left(\frac{||\nabla f(x)||_1}{\delta})^2\right]\right.$;
    $x^{(k+1)}$=max$(0, x^{(k+1)})$; // Ensure image positivity
  end for
  Until stopping criteria are met
end while

However, only isotropic edge property was considered in the EPTV model. The anisotropic edge property of an image is considered in the adaptive-weighted TV (AwTV) model proposed by Liu *et al* in [80].

Before moving on to the AwTV model, the difference between the isotropic and anisotropic discretisation of TV is given here [53].

An isotropic discretisation of TV is

$$TV_{iso}(x) = \sum_{ij} ||D_{i,j}x||_2 := \sum_{ij} g_{i,j}\sqrt{|x_{i,j+1} - x_{i,j}|^2 + |x_{i+1,j} - x_{i,j}|^2} \tag{4.34}$$

Where $g_{i,j}$ are weights. The isotropic discretisation of TV assigns zero weight to pixels on edges to free them from the TV minimisation. By doing so, it is fine if both pairs of pixels, i.e. (i,j) $(i+1,j)$ and (i,j) $(i,j+1)$ are cut by edges. However, this is not a common case. A more often case is when one of the pairs of pixels cut by an edge and not both. In this case, freeing just one pair of pixels from TV minimisation is more desirable.

An anisotropic discretisation of TV is

$$TV_{aniso}(x) = \sum_{\alpha} g_\alpha |D_\alpha x| := \sum g_{(i,j)\sim(k,l)} |x_{i,j} - x_{k,l}| \tag{4.35}$$

Where $g_{(i,j)\sim(k,l)}$ are weights and the summation is taken over all pairs of neighbours (i,j) and (k,l).

Next, the AwTV model is presented. The associated weights in the AwTV model are expressed as an exponential function, which can be adaptively adjusted with the local image-intensity gradient for the purpose of preserving the edge details.

The AwTV model is defined as:

$$||x||_{AwTV} = \sum_{ij} \sqrt{w_{i,i-1,j,j}(x_{i,j} - x_{i-1,j})^2 + w_{i,i,j,j-1}(x_{i,j} - x_{i,j-1})^2} \tag{4.36}$$

$$w_{i,i-1,j,j} = exp\left[-\left(\frac{x_{i,j} - x_{i-1,j}}{\delta}\right)^2\right] \tag{4.37}$$

$$w_{i,i,j,j-1} = exp\left[-\left(\frac{x_{i,j} - x_{i,j-1}}{\delta}\right)^2\right] \tag{4.38}$$

where δ controls the strength of the diffusion during each iteration. It should also be noted that the , in equation 4.36 are used to separate indices and not for differentiation. This anistropic TV type regularisation is based on the framework proposed by Perona and Malik [102], which was proved successful for X-ray tomography in [131].

With the AwTV model, it is possible to fully consider the gradient of the image and incorporate changes of local voxel intensities. This makes the TV minimisation by using the AwTV model more adaptive to the local information of the image. When the change in local voxel intensities are small, a stronger weight maybe given to emphasize the TV minimisation in the non-edge region. On the other hand, when there is a large change in the local voxel intensities, a weaker weight maybe given to alleviate the effect of TV minimisation in the possible edge regions of the image. Intuitively, the AwTV approaches the conventional TV norm in 4.5 when the weight goes to 1. Hence, the TV model is considered as a special case of the AwTV model when $\delta \to \infty$

Liu *et al* in [80] also proposed the algorithm called the AwTV-POCS algorithm, which minimises the AwTV subject to data and non-negativity constraints in the equations 4.7,4.8, for the purpose of dose reduction via CT image reconstruction from sparse data.

In the AwTV-POCS algorithm, similar two-phase approach as that of the ASD-POCS algorithm is employed. The pseudocode of the AwTV-POCS algorithm is shown in Algorithm 10. In the AwTV-POCS algorithm, the AwTV of the reconstructed image is minimised instead of the conventional TV norm. By doing so, the edge information of the reconstructed image can be preserved and the over-smoothing problem when using the conventional TV is mitigated.

Apart from the over-smoothing problem of implementing TV regularised algorithms, another great concern is the difficulty of selecting the hyper-parameters. Typically, an objective function of regularisation algorithms contains several terms which are specifically designed for different purposes. The balance between each term is controlled by hyper-parameters and is significant for the efficacy of the algorithms. The interactions between each hyper-parameter are complex and effective configurations of these hyper-parameters are difficult to identify. Generally, the choices of hyper-parameters are obtained manually through trials-and-errors as can be seen in many studies [64], [122], [65], [23], [66]. This is particularly true for the case of TV regularisation algorithms. One important part of this thesis is the sensitivity analysis of the hyper-parameters for some common TV regularisation algorithms. The analysis will be presented and discussed in more detail next chapter.

Algorithm 10: Adaptive-weighted Total Variation Projection Onto Convex Sets (AwTV-POCS) algorithm

Initialisation ;
Set δ,ε,τ,β,ng ;
$\vec{x} = 1$;
End of initialisation step;
while *Stopping criteria not met* **do**
  $\vec{x}_{prev} = \vec{x}$;
  for n_{angles} **do**
    $\vec{x} = \vec{x} + \beta V^{-1} A^T W(\vec{b} - A\vec{x})$; // SART update
  end for
  $\beta = \beta \times \beta_{red}$;
  $\vec{x}$=max$(0,\vec{x})$; // Enforce positivity constraint
  $\vec{x}_{out} = \vec{x}$;
  $\varepsilon_{current} = ||A\vec{x} - \vec{b}||$;
  $d_{projection} = ||A\vec{x} - A\vec{x}_{prev}||_2$;
  $dp = ||\vec{x} - \vec{x}_{prev}||_2$;
  $w_{i,i-1,j,j} = exp\left[-\left(\frac{\vec{x}_{ij} - \vec{x}_{i-1j}}{\delta}\right)^2\right]$;
  $w_{i,i,j,j-1} = exp\left[-\left(\frac{\vec{x}_{ij} - \vec{x}_{ij-1}}{\delta}\right)^2\right]$;
  $\vec{x}_{prev} = \vec{x}$;

  for *number of sub-iteration=1:ng* **do**
    $\vec{dx} = \nabla_{\vec{x}} ||\vec{x}||_{AwTV}$;
    $\hat{dx} = \frac{\vec{dx}}{||\vec{dx}||}$;
    $\vec{x} = \vec{x} - dp \cdot \tau \cdot \hat{dx}$; // AwTV minimisation update
  end for
  $dg = ||\vec{x} - \vec{x}_{prev}||$; // Change in TV update

  if $d_{projection} < \varepsilon$ **then**
    $\varepsilon = 0.995 \times \varepsilon$
  end if
  $\tau = \tau \times 0.995$;
  $cos\theta = \frac{\vec{dp} \cdot \vec{dg}}{||\vec{dp}|| \cdot ||\vec{dg}||}$;
  if $(cos\theta < -0.9$ ***and*** $\varepsilon_{current} < \varepsilon)$ ***or*** $\beta < 0.0005$ **then**
    Stop ; // Check stopping criteria
  end if
  Until stopping criteria are met
end while

4.7 Projection-Controlled Steepest Descent (PCSD)

There are several adaptations and improvements based on the same methematical approach as the ASD-POCS algorithm in the literature such as [79], [80], [125], [139]. One of these improvements is the PCSD algorithm, which is proposed by Liu et al [78]. The main objective of this algorithm is to determine the step-sizes of steepest descent method in the TV minimisation phase adaptively by considering the changes in projection domain of the POCS update, while the step size of ART operation in POCS is determined based on the data noise level. In addition, the PCSD algorithm implements a mechanism to reduce a computational cost by skipping the implementation of ART when the projection error of the current iteration is not greater than a predefined error bound. Algorithm 11 shows the pseudo code of the PCSD algorithm.

Algorithm 11: Projection-Controlled Steepest Descent (PCSD) algorithm

Initialisation ;
Set β,ε,ng,w_t and $x^{(0)}$;
Set $w = 0$, $\eta = 1$, $k = 1$;
End of initialisation step;
while *Stopping criteria not met* **do**
  for *w=1:* w_t **do**
    $x_{ART} = x^{(w)}$;
    $\nabla_p^{(w)} = ||Ax_{ART} - b||$;
    if $((\nabla_p^{(w)})^2 > \varepsilon)$ **then**
      $x_{ART} + \frac{\beta_i}{||a_i||^2}(b_i - a_i x_{ART})a_i^T$; // ART update
    end if
    x_{POCS}=max$(x_{ART}, 0)$; // Enforce positivity constraint
    $n = 0$; // Initialisation of TV sub-iteration
    $x_{TV}^{(0)} = x_{POCS}$;
    if $(w > 0$ *and* $(\nabla_p^{(1)})^2 > \varepsilon)$ **then**
      $\eta = k\left(\frac{\nabla_p^{(w)}}{\nabla_p^{(1)}}\right)$;
    end if
    $s^{(n)} = \left(\frac{\partial ||x^{(n)}||_{TV}}{\partial x^{(n)}}\right)$;
    for *number of sub-iteration=1:ng* **do**
      $x_{TV}^{n+1} = x_{TV}^{n} - \eta \times \frac{s^{(n)}}{||s^{(n)}||}$; // TV minimisation update
    end for
  end for
  $x^{(w)} = x_{TV}^{n_t}$;
  Until stopping criteria are met
end while

As mentioned earlier, the iterative PCSD algorithm follows two-phase approach of the ASD-POCS algorithm, which consists of POCS and TV minimisation steps. Two phases of operations

are implemented alternately until the currently estimated image satisfies the stopping criteria. These criteria are (1) the image satisfies the constraints (2) pre-specified number of iteration is met (3) a difference between images of two consecutive iteration reaches the pre-defined small value.

In the PCSD algorithm, the step size of the TV minimisation phase is determined automatically by following this equation:

$$\eta = k\Big(\frac{\nabla_p^{(w)}}{\nabla_p^{(1)}}\Big) \tag{4.39}$$

where $\nabla_p^{(w)} = ||Ax^{(w-1)} - b||$ is the estimation error in the projection domain at the wth main iteration, $\nabla_p^{(1)}$ is the projection error after the first iteration.

It is worth mentioning that the square root function of the TV norm in equation 4.5 is non-differentiable at zero, which precludes the use of conventional gradient-based methods for optimisation [107]. In order to prevent this non-differentiability, the square root is modified by adding a small constant at the end.

By implementing the PCSD algorithm, the number of critical hyper-parameters to be calibrated and the computational cost of running the algorithm can be reduced.

4.8 Adaptive-weighted Projection-Controlled Steepest Descent (AwPCSD)

With the knowledge of the advantages and disadvantages of the TV regularisation algorithms we have learned so far, a completely new approach namely Adaptive-weighted Projection-Controlled Steepest Descent (AwPCSD) algorithm is introduced and tested in this section. The AwPCSD algorithm is developed to reconstruct the volumetric image from cone-beam CT projection data. The main algorithm of the AwPCSD is inspired from the PCSD algorithm by having two-phase approach with automatic computation of step size in TV minimisation phase. In addition to that, instead of minimising the conventional TV norm of the image, the AwPCSD algorithm minimise the AwTV norm instead. The AwTV norm that is used for the AwPCSD algorithm is extended from equation 4.36 for the 3D case as shown in equation 4.40.

$$||x||_{AwTV} =$$

$$\sum_{ijk} \sqrt{w_{i,i-1,j,j,k,k}(x_{i,j,k} - x_{i-1,j,k})^2 + w_{i,i,j,j-1,k,k}(x_{i,j,k} - x_{i,j-1,k})^2 + w_{i,i,j,j,k,k-1}(x_{i,j,k} - x_{i,j,k-1})^2} \tag{4.40}$$

$$w_{i,i-1,j,j,k,k} = exp\left[-\left(\frac{x_{i,j,k} - x_{i-1,j,k}}{\delta}\right)^2\right] \tag{4.41}$$

$$w_{i,i,j,j-1,k,k} = exp\left[-\left(\frac{x_{i,j,k} - x_{i,j-1,k}}{\delta}\right)^2\right] \tag{4.42}$$

$$w_{i,i,j,j,k,k-1} = exp\left[-\left(\frac{x_{i,j,k} - x_{i,j,k-1}}{\delta}\right)^2\right] \tag{4.43}$$

With this framework, the AwPCSD algorithm is able to preserve the edges of the reconstructed image better with less critical hyper-parameters to calibrate. To summarise, the pseudo code for the AwPCSD algorithm is shown in Algorithm 12. Note that the SART algorithm is used for the data fidelity constraint, instead of the ART algorithm as in the original PCSD algorithm, to improve the rate of convergence.

4.8.1 Numerical Study of the AwPCSD algorithm

A numerical study is demonstrated in this section to show the performance of the AwPCSD algorithm. Projection images from digital 4D Extended Cardiac-Torso (XCAT) phantom [115] is used as a data set. Figure 4-2 shows the cross-sectional slices of the XCAT phantom. Poisson and Gaussian noise [135], [84] is added to the input projection data to simulate a realistic noise. The default noise is the combination of Poisson noise with maximum photon count of 60,000 and the Gaussian noise with mean and standard deviation of 0 and 0.5, respectively.

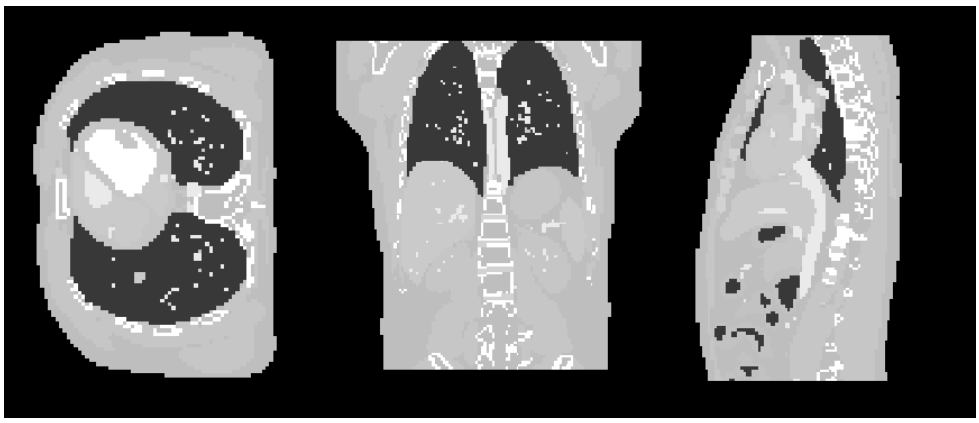


Figure 4-2: Cross-sectional slices of the XCAT phantom in transverse, coronal and sagittal planes.

In this experiment, CBCT data are generated consisting of 50 projection images, equally sampled over 360° circle. The reconstruction from the proposed AwPCSD algorithm is compared with FDK and iterative algorithms: CGLS and OS-SART, as well as other 3 TV regularisation algorithms: ASD-POCS, PCSD and AwTV-POCS. Four algorithms (FDK, OS-SART, CGLS, ASD-POCS) have already been implemented and are available in the TIGRE toolbox. The AwTV-POCS, PCSD and AwPCSD algorithms are developed by following the principles of each algorithm as explained in this chapter and employ GPU accelerated Ax and $A^T b$ operations from the TIGRE toolbox to speed up the execution. Cross-sectional slices of the reconstructed images from 7 reconstruction algorithms are shown in figure 4-3, in comparison with the exact phantom image.

Algorithm 12: Adaptive-weighted Projection-Controlled Steepest Descent (AwPCSD)

Initialisation;
Set x^0,β, β_{red}, ε, ng, δ;
Set $w = 0$, $\eta = 1$, $k = 1$;
End of initialisation step;
while *Stopping criteria not met* **do**
  for *w=1: w_t* **do**
    $x_{SART} = x^{(w)}$;
    $\nabla_p^{(w)} = ||Ax_{SART} - b||$;
    if *($(\nabla_p^{(w)})^2 > \varepsilon$)* **then**
      for n_{angles} **do**
        $x_{SART} = x_{SART} + \beta V^{-1} A^T W(\vec{b} - A\vec{x})$; // SART update
      end for
    end if
    x_{POCS}=max$(0, x_{SART})$; // Enforce positivity constraint
    $\beta = \beta \times \beta_{red}$;

    $$w_{i,i-1,j,j,k,k} = exp\left[-\left(\frac{x_{ijk}-x_{i-1jk}}{\delta}\right)^2\right];$$
    $$w_{i,i,j,j-1,k,k} = exp\left[-\left(\frac{x_{ijk}-x_{ij-1k}}{\delta}\right)^2\right];$$
    $$w_{i,i,j,j,k,k-1} = exp\left[-\left(\frac{x_{ijk}-x_{ijk-1}}{\delta}\right)^2\right];$$

    $n = 0$; // Initialisation of AwTV sub-iteration
    $x_{AwTV}^{(0)} = x_{POCS}$;
    if *($w > 0$ and $(\nabla_p^{(1)})^2 > \varepsilon$)* **then**
      $\eta = k\left(\frac{\nabla_p^{(w)}}{\nabla_p^{(1)}}\right)$;
    end if
    $s^{(n)} = \left(\frac{\partial ||x^{(n)}||_{AwTV}}{\partial x^{(n)}}\right)$;
    for *number of sub-iteration=1:ng* **do**
      $x_{AwTV}^{n+1} = x_{AwTV}^{n} - \eta \times \frac{s^{(n)}}{||s^{(n)}||}$; // AwTV minimisation update
    end for

  end for
  $x^{(w)} = x_{AwTV}^{n_t}$;
  Until stopping criteria are met
end while

The normalised root mean squared errors (NRMSE) of reconstructed images from all algorithms are also shown in the figure for quantitative comparison. The NRMSE is calculated by using the following equation [117].

$$NRMSE = \frac{RMSE}{f(x_i)_{max} - f(x_i)_{min}} \tag{4.44}$$

Where $f(x_i)_{max}$ and $f(x_i)_{min}$ denote the maximum and minimum intensity value of an image, respectively. The root mean squared error (RMSE) calculates the similarity between the resulting image and the reference image. The RMSE is defined as follows [133]:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{f}(x_i) - f(x_i))^2} \tag{4.45}$$

Where $\hat{f}(x_i)$ represents the reference attenuation coefficient at voxel i , $f(x_i)$ represents the reconstructed attenuation coefficients at voxel i, N is total number of voxels of the image. A small value of RMSE indicates small difference between the two images and vice versa.

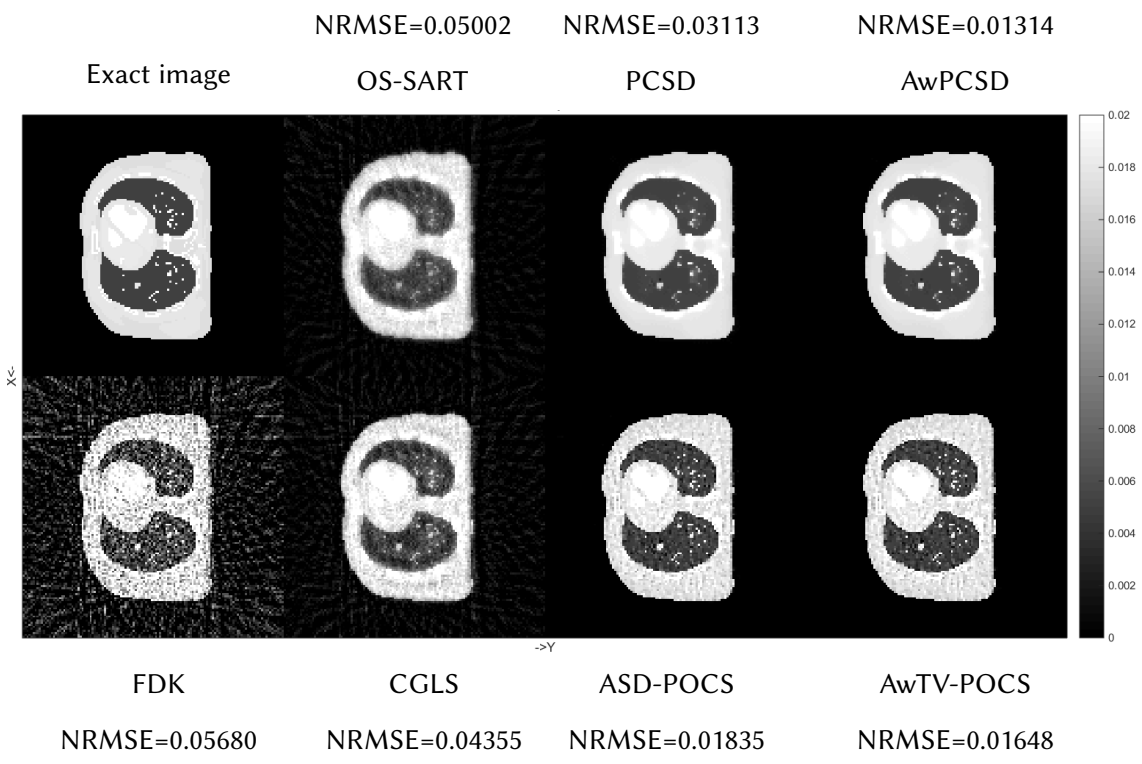


Figure 4-3: Cross-sectional slices in transverse plane of the reconstructed images from 7 reconstruction algorithms, in comparison with the exact phantom image. The normalised root mean squared errors (NRMSE) of all the images are also presented.

The details of implementation for each iterative algorithm are as follows. The CGLS algorithm converges at 15 iterations and 60 iterations are specified for OS-SART algorithm. All the TV regularisation algorithms are specified maximum number of iteration as 50 and default hyper-parameter values from the TIGRE toolbox. The maximum number of iteration of 50 is

specified as empirically found from trials-and-errors to represent a good amount of iterations needed to produce good quality of image for this experiment. The maximum number of iteration is used as one of the stopping criterion of the TV algorithms. In case that the algorithms are not stopped from satisfying the other stopping criterion, they will implement until the maximum number of iteration is reached. More details of the stopping criterion will be discussed in the section 5.1.2 of next chapter. This number is also served as a constraint of time allowed for the reconstruction to take place, for a comparison purpose. In this experiment, the ASD-POCS and AwTV-POCS algorithms converge and stop at 14 and 16 iterations, respectively. The PCSD and AwPCSD algorithms stop at the specified maximum number of iteration. Table 4.1 presents the reconstruction times of all the iterative algorithms in this experiment, as implemented using the computer with an Intel Core i7-4930K CPU@3.40GHz with 32 GB RAM and GPU: NVDIA GeForce GT 610.

Table 4.1: Reconstruction times of 6 iterative algorithms

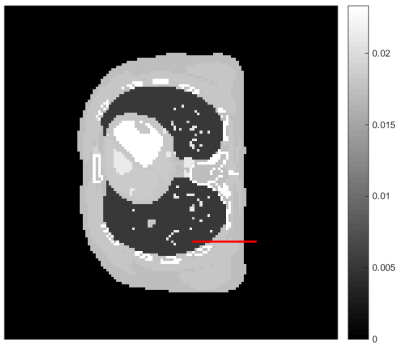
Algorithms	**Reconstruction time**
CGLS	29.3 seconds
OS-SART	36.6 seconds
ASD-POCS	1 minute, 57.3 seconds
AwTV-POCS	2 minutes, 27.6 seconds
PCSD	2 minutes, 16.6 seconds
AwPCSD	2 minutes, 35.6 seconds

According to figure 4-3 and table 4.1, the following observations can be seen:

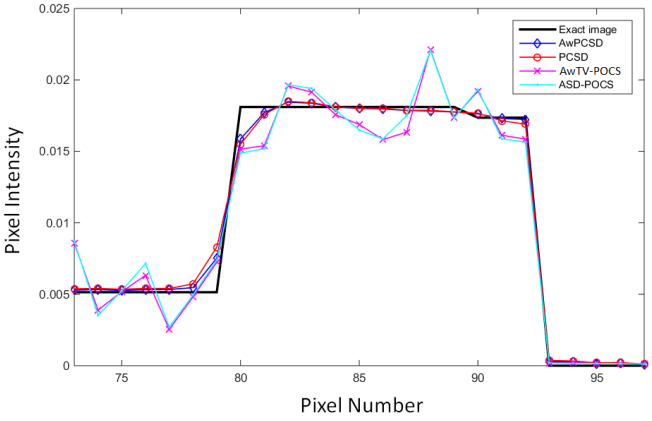
- Comparing between FDK and iterative algorithms (CGLS, OS-SART, ASD-POCS, AwTV-POCS, PCSD, AwPCSD), the results from the iterative algorithms contain much less artifacts than that of the FDK algorithm.
- Comparing between iterative algorithms (CGLS, OS-SART) and TV regularisation algorithms (ASD-POCS, AwTV-POCS, PCSD, AwPCSD), the results from the TV regularisation algorithms contain much sharper edges than those of the CGLS and OS-SART algorithms, as measured by lower NRMSE values presented in figure 4-3. However, the reconstruction times of the TV algorithms are longer, as shown in the table 4.1. Thus, there is a trade-off between the reconstruction quality and time.
- The reconstructed results from the ASD-POCS and AwTV-POCS algorithms are more noisy than the results from the PCSD and AwPCSD algorithms.

The results from 4 TV regularisation algorithms is analysed further by considering the one-dimensional profiles plot as shown in figure 4-4.

From figure 4-4, one-dimensional profiles from PCSD and AwPCSD are closer to the profile of the exact phantom image than those of the ASD-POCS and AwTV-POCS algorithms.



(a)



(b)

Figure 4-4: The intensities of the reconstructed images from the ASD-POCS, AwTV-POCS, PCSD, AwPCSD algorithms along the red line as shown in (a) are plotted in (b).

This means that the PCSD and AwPCSD algorithms can preserve the important features of the reconstructed image better. The differences between the adaptive-weighted and non-adaptive weighted algorithms are not clearly visible here. The comparison between the adaptive-weighted algorithms and the non-adaptive-weighted algorithms will be shown and discussed in more detail next chapter, together with the analysis of sensitivity of TV hyper-parameters.

4.9 Summary

In this chapter, the algorithms to solve CT reconstruction problem by minimising the total variation norm of the image as a regularisation are presented. The pioneering work such as the ASD-POCS algorithm, as well as some adaptations and improvements of the ASD-POCS algorithm are discussed. Advantages and disadvantages of the existing algorithms are extensively studied, which leads to the development of the novel algorithm, the AwPCSD algorithm, as a part of the contributions of this thesis. It is clearly shown in the numerical study that the results from the TV regularisation algorithms are superior to those of the iterative algorithms such as CGLS and OS-SART and much better than the FDK result, in a limited data scenario. However, one of the main drawbacks of the TV regularised algorithms is its reliance on the selection of hyper-parameters, which are used to balance the effects between the data fidelity and the TV minimisation processes. The in-depth analysis of the sensitivity of each hyper-parameter will be discussed in the next chapter.

Chapter 5

Sensitivity analysis of TV hyper-parameters

The iterative reconstruction algorithms, especially with regularisation techniques, are based on numerical optimisation. The balance between the constraints and objective function is controlled by a diverse set of hyper-parameters. As briefly mentioned in the previous chapter, one of the reasons behind the complications of implementing the TV regularisation algorithms is the selection of TV hyper-parameters. The TV regularisation algorithms require careful selection of these hyper-parameters for which there is no well established criteria. This chapter presents a comprehensive valuation of hyper-parameter selection in 4 common TV-based reconstruction algorithms comprising ASD-POCS, AwTV-POCS, PCSD and the newly proposed AwPCSD. The aim is to analyse the sensitivity of the hyper-parameters required for the TV regularisation algorithms. These hyper-parameters play an important role in the reconstruction performance of the algorithms. It is useful to be able to examine the sensitivity that the reconstruction image has to the changes in value of these hyper-parameters, in order to know which ones to prioritise when tuning the algorithm. Ultimately, heuristics on how to choose these hyper-parameters are desired to minimise or completely avoid rerunning the reconstruction with different values of hyper-parameters.

The sensitivity of all the hyper-parameters is analysed using two image quality metrics: the root mean squared error (RMSE) and the correlation coefficient (CC) for the algorithm introduced in the previous chapter, AwPCSD, in comparison with other three existing algorithms, ASD-POCS, AwTV-POCS and PCSD. By doing this, the performance of the new algorithm AwPCSD can also be evaluated.

5.1 Stopping criterion and hyper-parameters for TV regularisation algorithms

As explained in the previous chapter, the implementations of all 4 TV based reconstruction algorithms being studied in this chapter are based on two-phase strategy that alternates between

data-consistency and TV minimisation steps. A set of hyper-parameters is required to adjust the contributions or balance these two operations. There is no straightforward way to determine the optimal values of hyper-parameters other than trial-and-error tests. Also, when reconstructing different types of images, different hyper-parameters need to be empirically chosen [79]. This process is very time-consuming. Therefore, it is useful to examine the sensitivity that the reconstruction image has to value change on these parameters, in order to know which ones to prioritise when tuning the algorithm.

In this chapter, the sensitivity analysis of hyper-parameters is implemented by observing the impact on the reconstruction performance using simulation experiments on thorax medical phantom [115]. The values of each hyper-parameter are varied in a specified range. The results are demonstrated visually and quantitatively using two image quality metrics, RMSE and CC.

5.1.1 Hyper-parameters for TV-based reconstruction algorithms

Different algorithms require different numbers of hyper-parameter for its implementation. This section compiles explanations of all the hyper-parameters required to implement TV regularization algorithms.

- Data-inconsistency-tolerance parameter (ε)

  The first hyper-parameter specifies tolerance of the image to be reconstructed. The value of this hyper-parameter controls an impact level of potential data inconsistency on image reconstruction. It is defined as the maximum L_2 error to accept image as valid. This hyper-parameter is used as one of the checks for the stopping criterion which are discussed in detail later on in this section.

- TV sub-iteration number (ng)

  Next hyper-parameter is TV sub-iteration number (ng), which specifies how many times the TV minimisation process is performed in each iteration of the algorithm.

- TV hyperparameter (α)

  This hyper-parameter is used to convert the steepest-descent step-size from a fraction of a step-size to an absolute image distance on the first iteration in ASD-POCS and AwTV-POCS algorithms. This hyper-parameter is not required in PCSD and AwPCSD algorithms as these two algorithms adaptively adjust the step-size of steepest-descent according to the difference from POCS update in the projection domain.

- Reduction factor of TV hyperparameter (α_{red})

  In the same way as the α, the α_{red} is only required by ASD-POCS and AwTV-POCS algorithms. If the ratio of change in the image due to TV minimisation to change in the image due to SART is greater than maximum ratio of change by TV minimisation to change by SART (r_{max}) and the L_2 error of image in the current iteration is greater than ε, simultaneously, the gradient-descent step-size is reduced by α_{red}.

- Relaxation parameter (β)

  This is a relaxation parameter on which the SART operator depends. The value of hyper-parameter starts at 1.0 and slowly decreases to 0.0 depending on the hyper-parameter β_{red}.

- Reduction factor of relaxation parameter (β_{red})

  This hyper-parameter is used to reduce relaxation parameter (β) in the SART step as the iteration progresses. When relaxation parameter (β) reduces to less than 0.005, the algorithm will be stopped since no further difference is noticeable between the reconstructed images of the current and next iterations. This is used as one of the stopping criterion, which will be explained in the next section.

- Maximum ratio of change by TV minimization to change by SART (r_{max})

  As mentioned in the part of hyper-parameter α_{red}, if the ratio of the change in the image due to steepest descent to the change in the image due to POCS is greater than r_{max} , the gradient-descent step-size is reduced by α_{red}.

- Scale factor for adaptive-weighted TV norm (δ)

  This hyper-parameter is only required in adaptive-weighted algorithms, i.e. AwTV-POCS and AwPCSD. It controls the strength of the diffusion in the adaptive-weighted TV norm in equation 4.36 during each iteration, as explained in the previous chapter. The weights in equation 4.36 make it possible to consider the gradient of the desired image and to take into account the changes of local voxel intensities.

5.1.2 Stopping criterion

The stopping criterion play an important role in the reconstruction algorithms as they specify at which point the algorithms should be stopped. Generally speaking, the algorithms are stopped either when the difference of image between current and previous iterations is not noticeable, the image is accepted to be an optimal one or the maximum number of iteration is reached due to the time constraint. The stopping criterion utilised in the reconstruction algorithms in this chapter comprise of 3 checks.

Firstly, the case where reconstructed image is accepted to be an optimal solution is considered. When implementing the algorithm, the currently estimated image is checked to see if it obeys the constraints of equations 4.7 and 4.8, as explained in the previous chapter. In addition, the currently estimated image is checked if it satisfies the Karush Kuhn-Tucker conditions, which is a generalisation of the Lagrange multipliers for inequality constraints. The Karush Kuhn-Tucker conditions are presented in detail in section4.2 of chapter4. In summary, the KKT conditions state that the image will be an optimal solution if the TV and data-constraints gradients vectors are pointing in exactly the opposite direction. As a stopping criteria, these two vectors are checked if they are back-to-back with each other by computing the dot product between them. The Karush Kuhn-Tucker conditions are checked as one of the stopping criterion

to ensure that only the optimal solution is obtained regarding both the TV norm minimisation and data fidelity, even if the data constraint is satisfied. The reader is referred to read more detailed explanation of the neccessary conditions for a given image to be the optimal one in [122].

The first check of the stopping criterion is when the currently estimated image satisfies the following conditions:

$$c < -0.99 \quad \& \quad dd \leq \varepsilon \tag{5.1}$$

where c is the cosine of the angle between the TV and data-constraint gradients, dd is the L_2 error between the measured projections and the projections computed from the estimated image in the current iteration , ε is the data-inconsistency-tolerance parameter. Ideally, $cos\theta$ should be -1.0 to show that the two vectors are exactly opposite to each other, i.e. the angle between them is 180°. However, this value is rather difficult to achieve in practice and requires a large number of iterations. It is reported in [122] that ,empirically, there is only little significant changes in the image when the $cos\theta$ is below -0.5. Thus, the aim is to attain $cos\theta$ in the neighbourhood of -0.99.

In an ideal scenario, the value of c should be -1 as the vectors of TV and data-constraint gradients must be completely opposite to each other, as mentioned earlier. Practically, this value is difficult to reach as it requires a large number of iterations. Thus, this value is set to -0.99. The algorithm is stopped when the currently estimated image satisfies these two conditions in equation 5.1, simultaneously.

Secondly, the algorithms are ceased when the relaxation parameter (β) of the current iteration meets the following condition:

$$\beta < 0.005 \tag{5.2}$$

The relaxation parameter reduces every iteration by a factor of specified β_{red}. When value of β falls below 0.005, no further difference is noticeable between the reconstructed images of the current and next iterations. Hence, the algorithm is forced to stop.

Last check of the stopping criterion is when the maximum number of iteration specified by user is reached. As the number of iterations is a constant and cannot be updated adaptively, the algorithm may be stopped either before or after the optimal solution is obtained. The stopping criterion of maximum number of iteration can be related to the β condition in equation 5.2 depending on the specified values of β and β_{red}. For instance, when β and β_{red} are specified as 1 and 0.99, respectively, β will be reduced to $\beta = \beta \times \beta_{red}$ in every iteration of algorithm. In this case, it will take 528 iterations until $\beta < 0.005$ and satisfy the stopping criterion in equation 5.2. Thus, the stopping criterion of β also determines the maximum number of iteration.

When the algorithms meet any of these three stopping criterion checks, the current iteration is ceased and the currently estimated image is accepted to be a final reconstructed image.

5.2 Results

All 4 TV-based reconstruction algorithms are implemented based on the algorithms available in Tomographic Iterative GPU-based Reconstruction (TIGRE) toolbox [9]. Apart from analysing the sensitivity of TV hyper-parameters and evaluating the performance of the AwPCSD algorithm, we are also looking to investigate how the adaptive-weighted function helps to improve the result by comparing between two pairs of adaptive-weighted and non-adaptive-weighted algorithms.

The data set used to evaluate the performance of reconstruction algorithms and hyper-parameters analysis is a digital XCAT phantom which contains thorax anatomy structures [115]. The Poisson and Gaussian noise [135], [84] has been added to the input projection data for a simulation of realistic noise. The default noise is the combination of Poisson noise with maximum photon count of 60,000 and the Gaussian noise with mean and standard deviation of 0 and 0.5, respectively. Figures [5-1a] and [5-1b] show one cross-sectional slice of the exact thorax phantom and the region of interest used to compute the evaluation metrics, respectively.

(a) Exact phantom

(b) Region of interest

Figure 5-1: One cross-sectional slice of thorax phantom data set

Number of projection images used for image reconstruction in this study is 20 views, equally sampled from 360°. For this situation with very limited number of projection images, reconstruction using standard FDK algorithm can give an extremely poor quality reconstructed image with streak artifacts as shown in the figure 5-2.

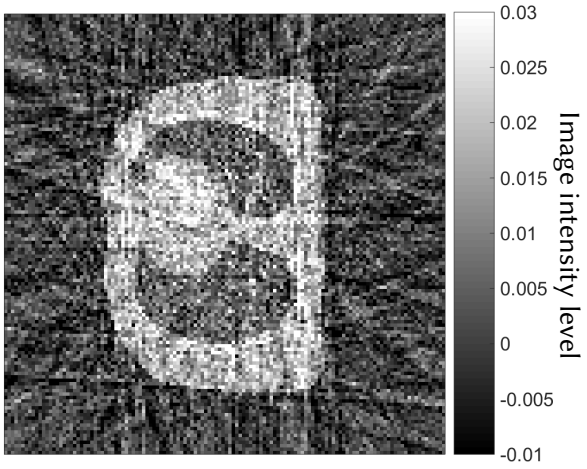


Figure 5-2: Reconstructed images from 20 projection images using FDK method

With this limited data scenario, the performance of the proposed AwPCSD algorithm as well as other three existing TV-based reconstruction algorithms can be intensively evaluated and compared. The following two metrics were used as a quantitative measure of reconstruction quality in this study.

5.2.1 Image Quality Metrics

- Root mean squared error (RMSE)

  The first metric is root mean squared error (RMSE) which calculates the similarity between the resulting image and the reference image. The RMSE is defined as follows [133]:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{f}(x_i) - f(x_i))^2} \tag{5.3}$$

  where $\hat{f}(x_i)$ represents the reference attenuation coefficient at voxel i , $f(x_i)$ represents the reconstructed attenuation coefficients at voxel i, N is total number of voxels of the image. A small value of RMSE indicates small difference between the two images and vice versa. In this experiment, the reference image is the exact XCAT phantom. The reference image is only used for assessment to determine how well the reconstruction algorithms are able to reconstruct an image, in comparison with the ideal case. In real practice, there is no reference image since a reconstruction is implemented on a measured projection data of an object and the image of the object is what is being produced. If an assessment is needed for a reconstruction with real measured data, a reference image could be a reconstructed image using a standard reconstruction algorithm such as the FDK algorithm using as many projections as possible. The reconstructed image from such scenario should be sufficient to represent the best possible expectation of the image to be reconstructed.

- Correlation coefficient (CC)

The second metric is correlation coefficient (CC) which measures the degree to which the two images are associated. The CC metric is defined as follows

$$CC = \frac{Cov(\hat{f}(x), f(x))}{\sigma_{\hat{f}(x)}\sigma_{f(x)}} \tag{5.4}$$

where $Cov(\hat{f}(x), f(x))$ is the covariance of the reference and reconstructed images, $\sigma_{\hat{f}(x)}$ is the standard deviation of the reference image, $\sigma_{f(x)}$ is the standard deviation of the reconstructed image. The value of CC is between -1 and 1 where 1 is the total positive linear correlation, 0 is no linear correlation and -1 is total negative linear correlation. A positive linear correction between the reference and reconstructed images is desired.

5.2.2 Sensitivity analysis of hyper-parameters

This section analyses a sensitivity of all the hyper-parameters required for 4 reconstruction algorithms: ASD-POCS, AwTV-POCS, PCSD and AwPCSD. Each hyper-parameter is analysed separately across all the algorithms. The RMSE and CC values calculated from final estimated images from each value of hyper-parameter are plotted to see the performance of 4 algorithms in respond to changing of hyper-parameters' values.

The total number of hyper-parameters is different among the 4 algorithms in this study. The table 5.1 shows the set of initial values for hyper-parameters that is used as a starting point to analyse the sensitivity of the first hyper-parameter, ε. The initial values in table 5.1 are chosen following some initial tests and also randomly selected.

Table 5.1: A set of initial values of hyper-parameters for the sensitivity analysis

Algorithms	ng	α	α_{red}	β	β_{red}	r_{max}	δ
ASD-POCS	25	0.002	0.95	1	0.98	0.94	N/A
AwTV-POCS	25	0.002	0.95	1	0.98	0.94	0.001
PCSD	10	N/A	N/A	1	0.98	N/A	N/A
AwPCSD	10	N/A	N/A	1	0.98	N/A	0.001

In this study, the ASD-POCS algorithm is used as a reference algorithm, which requires 7 hyper-parameters including ε. The AwTV-POCS requires one more hyper-parameter from ASD-POCS which is δ as it implements the adaptive-weighted TV norm, making AwTV-POCS the algorithm that requires the highest number of hyper-parameter among the 4 algorithms. The PCSD and AwPCSD algorithms require three less parameters than ASD-POCS and AwTV-POCS algorithms including α, α_{red} and r_{max}, making PCSD the algorithm that requires the least number of parameters among all 4 algorithms.

Reconstruction time

The reconstruction time of the 4 TV regularised algorithms in this analysis depends on several factors. The first factor is the experimental settings, which can be summarised for this analysis as following: Detector size = 512×512, Voxel size = $128 \times 128 \times 128$, Number of projections = 20 projections, maximum number of iteration = 500. The specifications of the testing computer is also important. In this experiment, the testing computer is an Intel Core i7-4930K CPU@3.40GHz with 32 GB RAM and GPU: NVDIA GeForce GT 610. If these 2 factors change, the reconstruction time will also change accordingly.

Moreover, different values of some hyper-parameters strongly affect the reconstruction time. These are the ones that directly affect the stopping criterion such as ε and β_{red}. Other hyper-parameters can also affect the reconstruction time such as ng since it defines a number of TV sub-iteration within the main algorithm.

Tables 5.2 and 5.3 report examples of different reconstruction times for the 4 TV regularised algorithms in this experiment. Some extreme values of hyper-parameters are chosen to give an idea on estimated time required for different values.

Table 5.2: Reconstruction times for 4 TV regularised algorithms with $\varepsilon = 0$ and $\varepsilon = 100,000$. The other hyper-parameters are defined as shown in table 5.1.

$\varepsilon = 0$		
Algorithms	**Time**	**Stopping criterion**
ASD-POCS	7 mins 55.3 secs	maximum iteration at 500
AwTV-POCS	9 mins 57 secs	maximum iteration at 500
PCSD	≈4 mins	β at iteration 263
AwPCSD	≈4 mins	β at iteration 263
$\varepsilon = 100,000$		
Algorithms	**Time**	**Stopping criterion**
ASD-POCS	≈40 secs	c and dd at iteration 21
AwTV-POCS	≈40 secs	c and dd at iteration 20
PCSD	≈4 mins	β at iteration 263
AwPCSD	≈4 mins	β at iteration 263

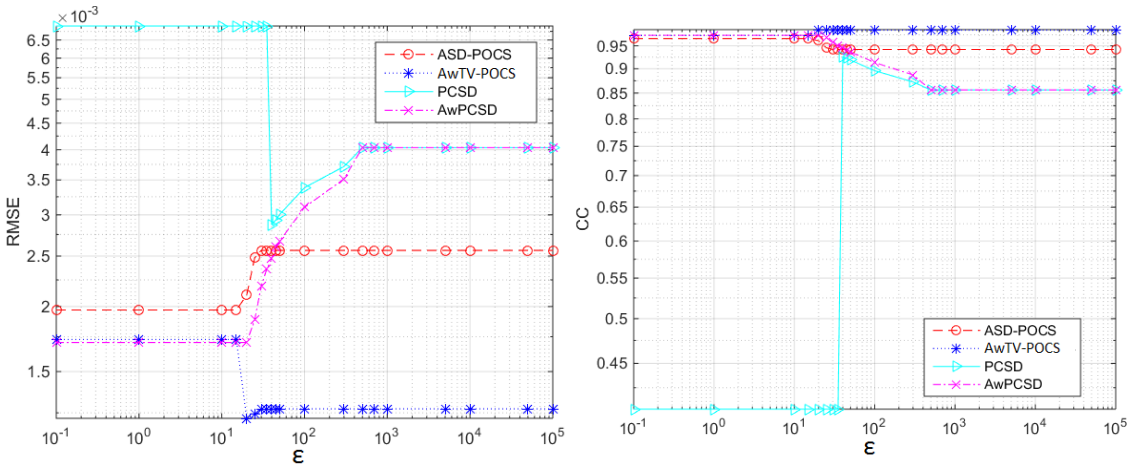
Table 5.3: Reconstruction times for 4 TV regularised algorithms with $ng = 1$ and $ng = 200$ with ε chosen as 40. The other hyper-parameters are defined as shown in table 5.1.

$ng = 1$		
Algorithms	**Time**	**Stopping criterion**
ASD-POCS	8 mins 42 secs	maximum iteration at 500
AwTV-POCS	9 mins 28.6 secs	maximum iteration at 500
PCSD	≈4 mins	β at iteration 263
AwPCSD	≈6 mins	β at iteration 263
$ng = 200$		
Algorithms	**Time**	**Stopping criterion**
ASD-POCS	13 mins 27.5 secs	maximum iteration at 500
AwTV-POCS	13 mins 55.9 secs	maximum iteration at 500
PCSD	≈6 mins	β at iteration 263
AwPCSD	≈8 mins	β at iteration 263

Data-inconsistency-tolerance parameter (ε)

The first hyper-parameter to be analysed is ε, which is required by all 4 algorithms and significant to the final reconstructed image as it involves in stopping criterion of equation 5.1.

To begin the sensitivity analysis, the other hyper-parameters are set to their initial settings as displayed in table 5.1, while ε are varied from 0 to 10^5. The 21 values of ε being tested in this experiments are ε =0,0.1,1,10,15,20,25,30,35,40,45,50,100,300,500,700,1,000,5,000,10,000,50,000 and 100,000. The RMSE and CC values are calculated using the reconstructed image obtained from each ε value and plotted as shown in figure 5-3.



(a) RMSE plots from varying ε

(b) CC plots from varying ε

Figure 5-3: RMSE and CC plots across different ε values

According to the figure 5-3, an evaluation of effects from varying ε by using RMSE and CC plots can not be done in a straightforward way because all the algorithms are implemented with all 3 stopping criterion in place. When specifying different values of ε to the algorithms, some of the algorithms with certain values of ε were stopped according to different stopping criterion, not just from achieving the specified ε. As a consequence, the reconstructed images do not directly represent the effects from different values of ε.

Instead, the stopping criterion of the 4 algorithms are suppressed for this experiment of varying ε. All the algorithms are forced to implement until they reach the iteration number 250. This number is chosen such that the algorithms iterate enough to achieve the specified ε, especially when the values are small. Moreover, instead of using the RMSE and CC metrics, this experiment of varying ε is evaluated by plotting the l_2 norm of the difference between a projection of the estimated image and the simulated projection data ,$||Ax-b||_2$, and TV norm of the estimated image, $||x||_{TV}$.

Firstly, the analysis of the results in this section start with the plots between l_2 norm error, i.e. $||Ax-b||_2$, and the varying ε, as shown in figure 5-4.

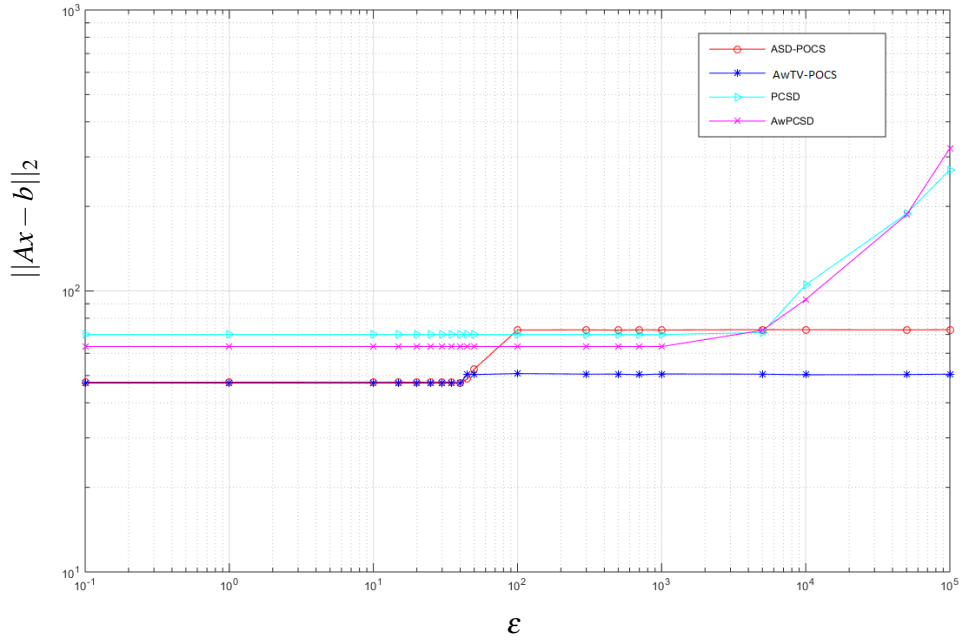


Figure 5-4: Plots of the l_2 norms of the difference between a projection of the estimated image and the simulated projection data ($||Ax-b||_2$) of the reconstructed images from 4 algorithms with varying values of specified ε from 0 to 10^5.

The overall of figure 5-4 can be interpreted that as the ε increase, the l_2 norm increases. For the ASD-POCS and the AwTV-POCS algorithms, l_2 norms stay at the same value at the beginning until they start to increase at $\varepsilon \approx 40-50$ and stay constant at different values afterwards. The l_2 norms of PCSD and AwPCSD algorithms start with slightly higher values and stay constant until they start to increase at the higher values of ε and keep increasing until the last value of ε.

According to the figure 5-4, it can also be seen that the l_2 norms measured from the final reconstructed images at the end of the implementation and the specified ε values are not the same values. In case that large ε values are specified, the final $||Ax-b||_2$ achieved in the reconstructed image is even lower than that of the specified value. For instance, the case of ε = 10^5, we can see that the final $||Ax-b||_2$ achieved by each algorithm ended up all lower than 10^5. This is because the algorithms are forced to iterate until the maximum number of iteration 250 is reached, which makes the algorithms still minimising the error even though the achieved error passes the point of the specified values.

However, when the ε is specified at very low values, such as 0.1 and 0, we can see that the final $||Ax-b||_2$ achieved by each algorithm can only get to the value $\approx 50-70$, even though the 250 iterations of algorithms are implemented. The point to say here is that very low ε are almost impossible to achieve and also the specified ε to the algorithms are not always guaranteed that the final l_2 norm error of the same value will be achieved.

Next, the plots between TV norm of the estimated image, $||x||_{TV}$ and the varying ε are shown in figure 5-5.

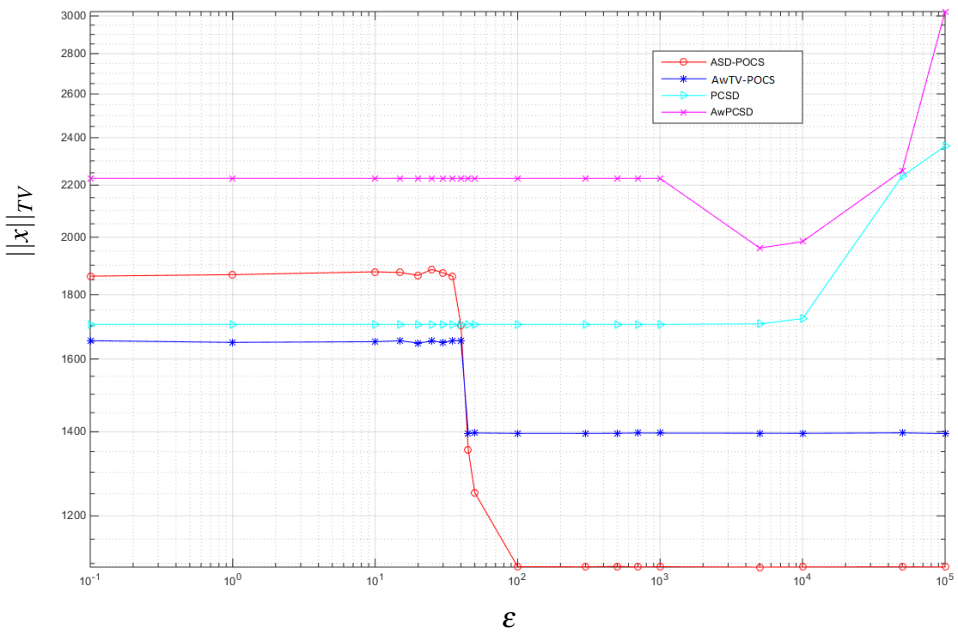


Figure 5-5: Plots of the TV norms of the reconstructed images from 4 algorithms with varying values of specified ε from 0 to 10^5.

According to figure 5-5, as the ε increases, the TV norms of the reconstructed images from ASD-POCS and AwTV-POCS algorithms stay constant at certain values upto $\varepsilon \approx 35-40$. Then, the TV norms of both algorithms drop and stay constant at the lower values afterwards until the last ε. For PCSD algorithm, the TV norms are constant with increasing ε at the beginning until they start to increase gradually at $\varepsilon \approx 5,000$. For AwPCSD algorithm, the TV norms also stay constant upto $\varepsilon \approx 1,000$, then they slightly decrease before turning to gradually increase at the point of $\varepsilon \approx 5,000$.

Next, we investigate how the 2 norms are changed, in relative to each other, with different values of specified ε. This is shown by plotting $||Ax-b||_2$ and $||x||_{TV}$ norms computed from the reconstructed images from 4 algorithms with different values of ε, as presented in figure 5-6.

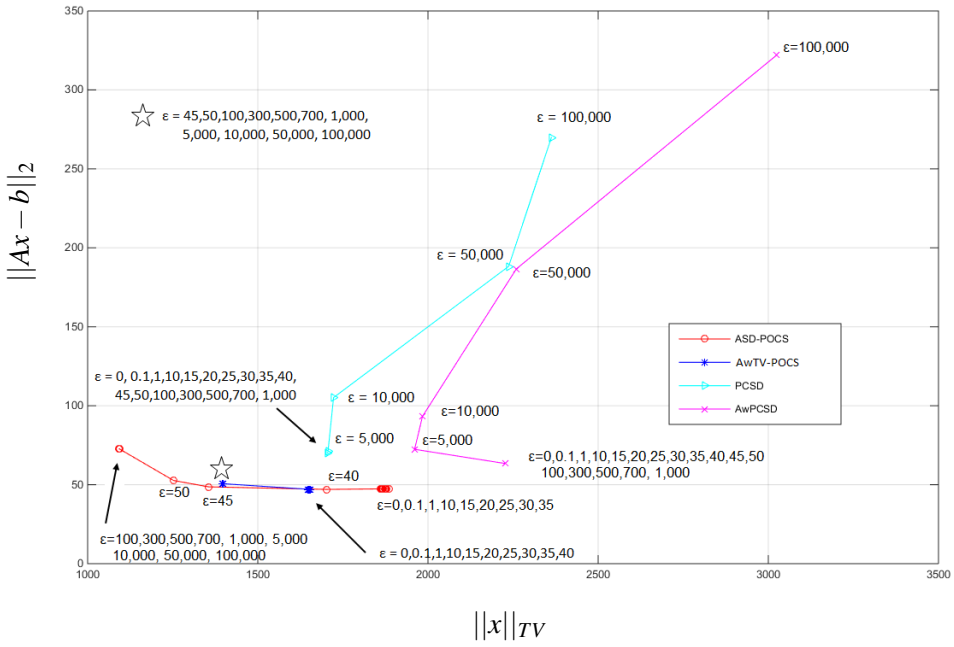


Figure 5-6: Plots of the $||Ax-b||_2$ and $||x||_{TV}$ norms of the reconstructed images from 4 algorithms with varying values of specified ε from 0 to 10^5. The Y axis represents $||Ax-b||_2$ and the X axis represents $||x||_{TV}$ with the values of ε annotated at each point.

According to figure 5-6, the results from the ASD-POCS and AwTV-POCS algorithms have the same pattern. As the ε decrease, the $||Ax-b||_2$ norms are also decreasing, as opposed to the increasing $||x||_{TV}$ norms. The PCSD and AwPCSD algorithms exhibit almost the same pattern with decreasing $||Ax-b||_2$ and $||x||_{TV}$ norms when the ε decrease. However, the $||x||_{TV}$ norm of the AwPCSD algorithm slightly increase when ε fall below 5,000.

As well as the plots of 2 norms with varying ε as presented in figures 5-4,5-5 and 5-6, the cross-sectional slices of reconstructed images from the experiment are shown in figure 5-7. The images from reconstruction using 4 values of ε are chosen to display in figure 5-7 out of the 21 values being tested, in order to give an information on how the reconstructed images are changing with an increasing of the ε.

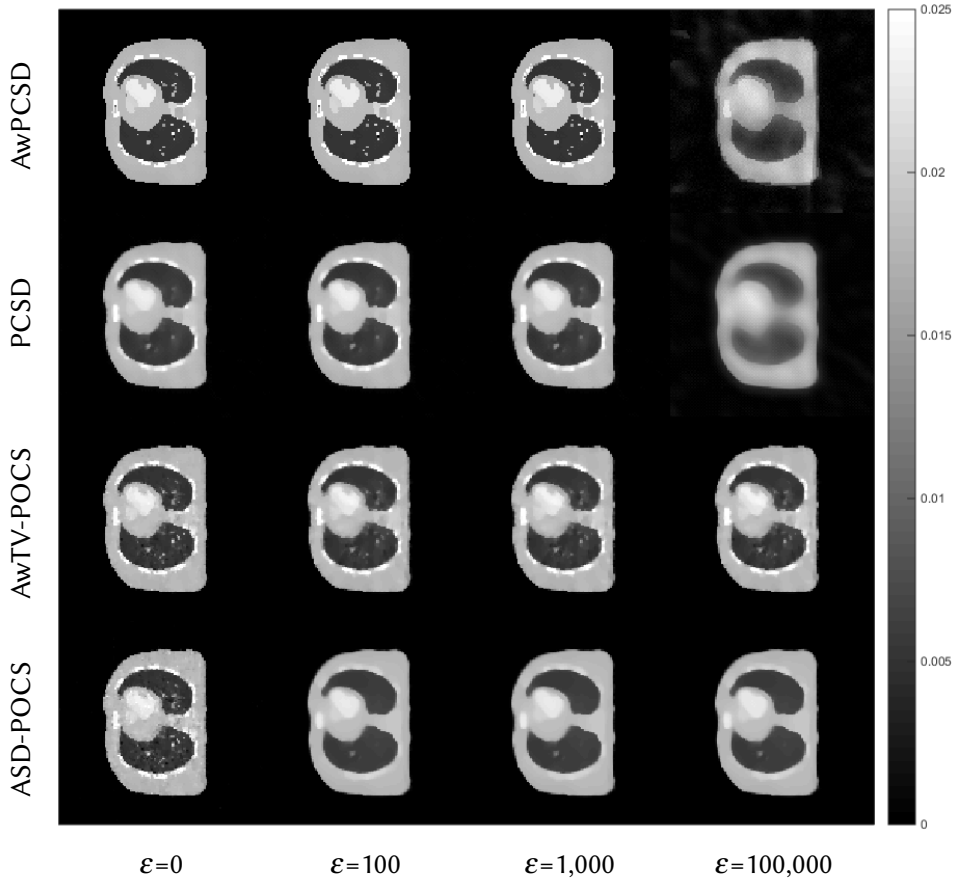


Figure 5-7: Reconstructed images from 4 algorithms using different ε values

With increasing values of ε, the quality of reconstructed images deteriorates. This can be seen from visually inspecting the images in figure 5-7, as well as the increasing of $||Ax-b||_2$ norm. However, the measured $||x||_{TV}$ norms do not always correspond to the quality of the reconstructed images. For instance, the $||x||_{TV}$ norms have shown to be decreasing with an increasing ε in the reconstructed images from ASD-POCS and AwTV-POCS algorithms. These are shown in flatter-looking images with less small features recovered in the 3rd and 4th rows from the top of figure 5-7, when the ε are getting larger. For the PCSD and AwPCSD algorithms, the reconstructed images with large ε deteriorates even more than those of the ASD-POCS and AwTV-POCS algorithms. These can be seen with the blurry edges with almost no small features of images recovered in the 1st and 2nd rows from the top of figure 5-7, when the ε are getting larger. However, the $||x||_{TV}$ norms have shown to be increasing with an increasing ε. These results show different amounts of image TV norm minimisation for each TV regularised algorithm. The main objective of CT reconstruction problem under study is to solve the constrained optimisation in the equation 4.6 explained in the previous chapter. The balance between a regularisation function, which is a minimisation of $||x||_{TV}$ norm, and the data fidelity constraint $||Ax-b||_2 < \varepsilon$ is controlled by a set of hyper-parameters. Each and every hyper-parameter is very important to the implementation of the TV regularised algorithms since it controls the balance of regularisation term and the data fidelity term. This is the main reason why the sensitivity analysis of

TV hyper-parameters in this chapter is very important.

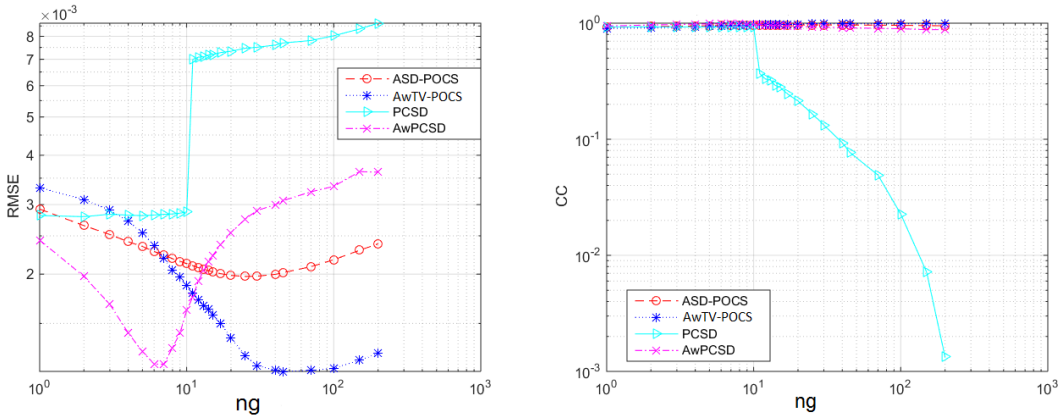
It is also worth noting that each TV regularised algorithm under study requires different numbers of hyper-parameters for its implementation. As much as the values of all the hyper-parameters strongly affect the final reconstruction results of all the algorithms, each hyper-parameter could also affects the others. However, the experiments in this study is designed to vary one hyper-parameter at a time while keeping others constant, in order to observe the effects of a reconstruction from one particular hyper-parameter with as little interference form other hyper-parameters as possible. The correlation between a group of hyper-parameters is a complicated issue and requires another study particularly devotes to this. We bear in mind that the results come out from this study are subject to this limitation.

It is very difficult to identify any specific value as the most appropriate ε for one particular data set as the ε is data-specific. However, from the trials and errors, we can suggest that using the L_2 norm of the reconstructed image obtained from Ordered-subset simultaneous algebraic reconstruction techniques (OS-SART) algorithm [19], [130] as an ε value yields an acceptable result. However, one has to bear in mind that specifying the ε is a matter of image quality versus computational time. Choosing too small ε might take longer computational time to achieve the desired value and also does not guarantee the best image for some of the algorithms. This can be seen from figure 5-3a that stopping criterion of reconstructed images with ε lower than 20 is from reaching the maximum number of iterations which takes long computational time. Moreover, in case of PCSD and AwTV-POCS algorithms, the reconstructed images from larger ε even have better quality than smaller RMSE values.

TV sub-iteration number (ng)

The second hyper-parameter to be analysed is TV sub-iteration number. The values are varied from 1 to 200. Other hyper-parameters are still the same as initial settings apart from ε, which are chosen to be the optimal values for each algorithm obtained from the previous section.

According to the RMSE plot in figure 5-8a, two algorithms which achieve the lowest RMSE values are AwTV-POCS and AwPCSD. Firstly, we consider the pair of PCSD and AwPCSD algorithms. For PCSD algorithm , the reconstructed images from low number of TV sub-iteration are still acceptable even though the images are not very clear as seen from the second row of figure 5-9. When the TV sub-iteration number increases, the RMSE values suddenly go up with relatively bad reconstructed images when $ng > 10$. This is to show that the strange performance of PCSD algorithm arises very easily, following on from sensitivity of ε to TV sub-iteration number in this case. However, with the adaptive-weighted function added on to PCSD algorithm, the AwPCSD algorithm can significantly improve the quality of reconstructed images as the TV sub-iteration number increases. The lowest RMSE value of the reconstructed images from AwPCSD is when $ng = 6$. After that, the reconstructed images get more blurred with the increasing TV sub-iteration number as seen in the first row of figure 5-9.



(a) RMSE plots from varying *ng*

(b) CC plots from varying *ng*

Figure 5-8: RMSE and CC plots across different *ng* values

Analysing the pair of ASD-POCS and AwTV-POCS algorithms, both algorithms improve the quality of reconstructed images as the TV sub-iteration number increases until they reach their lowest RMSE points which differ in both algorithms. For the ASD-POCS, the algorithm hits the lowest RMSE point when *ng* is approximately between 20 to 50. The AwTV-POCS algorithm is still able to bring down the RMSE further from the point of *ng* approximately equals to 6 until it reaches the lowest point when *ng* is between 30 to 100.

The two adaptive-weighted algorithms, AwTV-POCS and AwPCSD, can bring down the RMSE of the final reconstructed images to approximately the same level. According to figure 5-8a, the AwPCSD requires less TV sub-iteration number than AwTV-POCS to reach the optimal reconstructed image which can save significant amount of computational time. This is because the AwPCSD algorithm takes into account the current image error when choosing the TV steepest descent step-size for the next iteration. However, the AwPCSD is more sensitive to the changing values TV sub-iteration number as the quality of final reconstructed images deteriorate quickly with the increasing *ng* as shown in the figure 5-9.

To sum up, the adaptive-weighted algorithms outperform the non-adaptive-weighted algorithms over the range of TV sub-iteration numbers under study. Comparing between AwPCSD and AwTV-POCS, although the best reconstructed images from these 2 algorithms achieve the same level of RMSE, AwPCSD algorithm requires less number of *ng* which save computational time. However, it is important to specify a proper TV sub-iteration number when implementing AwPCSD algorithm as this hyper-parameter is very sensitive to values change.

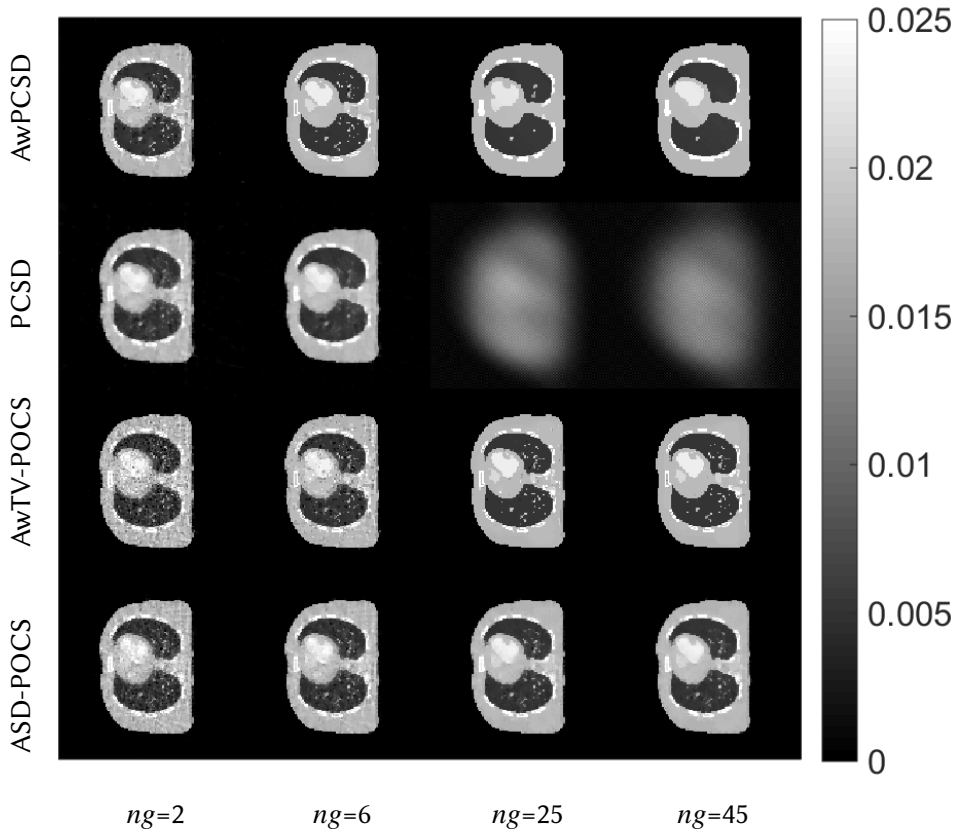
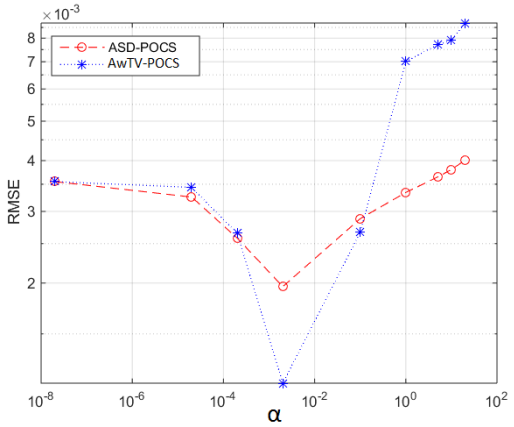


Figure 5-9: Reconstructed images from 4 algorithms with different *ng* values

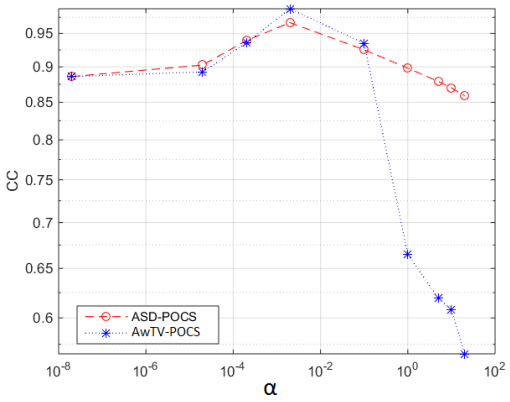
TV hyperparameter (α)

The next hyper-parameter to be studied is α. This hyper-parameter is only required by ASD-POCS and AwTV-POCS algorithms. In these 2 algorithms, the α is used to specify the steepest-descent step-size in the first iteration. As the iteration goes on, the TV step-size will be reduced by the amount of α_{red} if the condition which we briefly discussed earlier when explaining the data-inconsistency-tolerance ε hyper-parameter section is met.

The α hyper-parameter is varied from $0, 2 \times 10^{-8}, 2 \times 10^{-5}, 2 \times 10^{-4}, 2 \times 10^{-3}, 0.1, 1, 5, 10$ to 20. Other hyper-parameters are still the same as initial settings apart from ε and *ng* which are set as found in the previous sections. The RMSE and CC plots of final reconstructed images using different α values over the range of study are shown in figure 5-10.



(a) RMSE plots from varying α



(b) CC plots from varying α

Figure 5-10: RMSE and CC plots across different α values

From the RMSE plots, both algorithms have a minima in a very specific value of α ,i.e. when α = 0.002. The reconstructed images get increasingly deteriorated the further away the α is from that value. This can be seen from the reconstructed images in figure 5-12. Again, the AwTV-POCS algorithm has better performance than ASD-POCS algorithm with the clearer reconstructed image in the same value of α.

With the fixed value of $\alpha = 0.002$, figure 5-11 shows the RMSE plots of the reconstructed images using ASD-POCS and AwTV-POCS algorithms with varying TV sub-iteration number (ng) from 2 to 50.

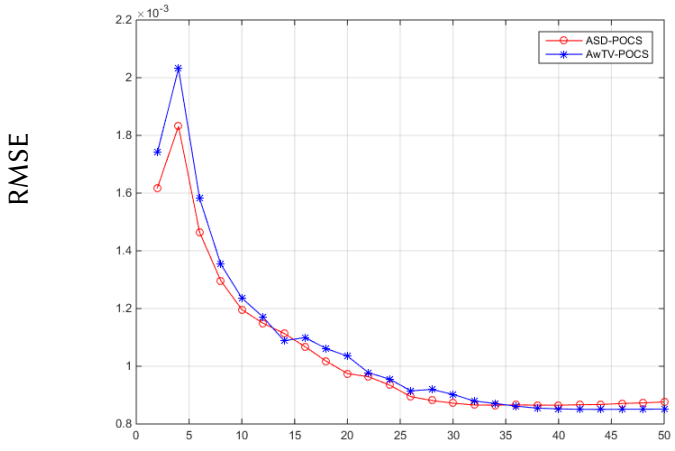


Figure 5-11: RMSE plots of the reconstructed images using ASD-POCS and AwTV-POCS algorithms with fixed value of $\alpha = 0.002$ and varying TV sub-iteration number (ng) from 2 to 50.

According to figure 5-11, it can be seen that the reconstructed images using ASD-POCS and AwTV-POCS algorithms have the lowest RMSE approximately at $ng = 35$.

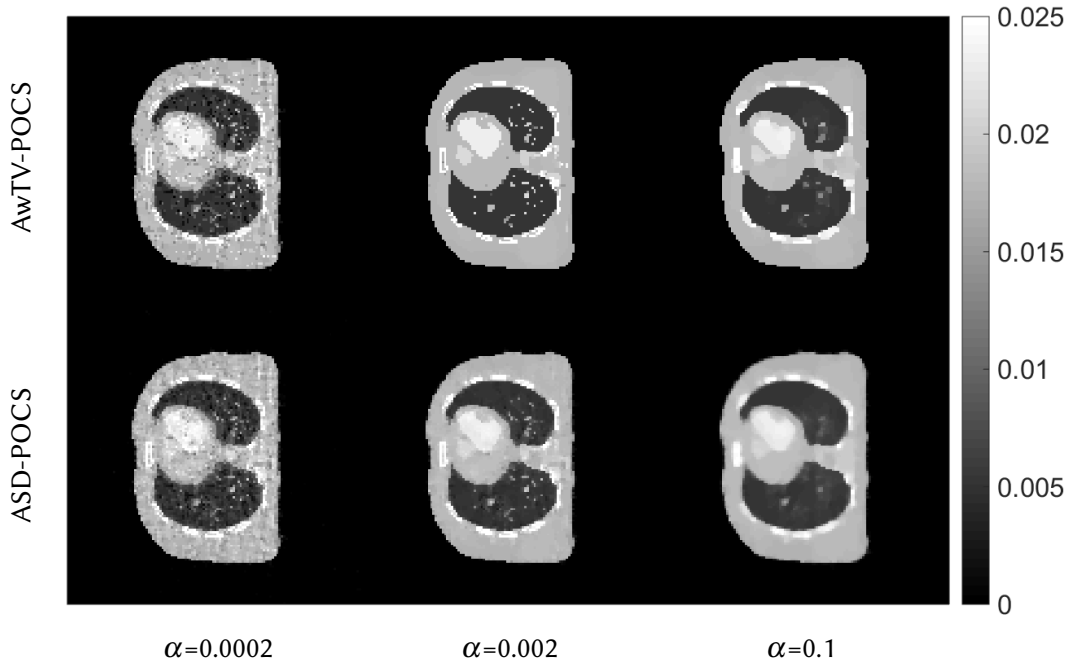


Figure 5-12: Reconstructed images from ASD-POCS and AwTV-POCS with different α values

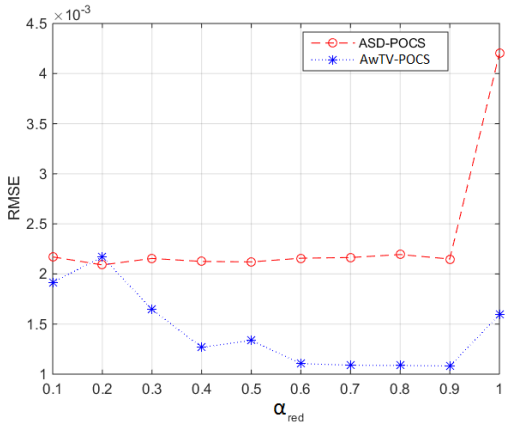
Therefore, it is of the utmost importance to specify a proper value of α for the ASD-POCS and AwTV-POCS algorithms to work properly. With this knowledge, we can say that this is a great advantage of PCSD and AwPCSD algorithms that do not require this α hyper-parameter

It is also worth noting that the hyper-parameter α is data set dependent. According to the experiments in [68], different values of α can lead to transition from an under-regularised noisy/patchy TV-regularised reconstruction, through to an over-regularised solution.

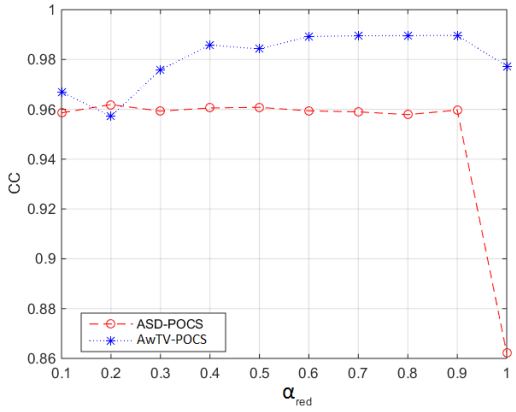
Reduction factor of TV hyperparameter (α_{red})

In the same way as α, the α_{red} hyper-parameter is only required by two algorithms, ASD-POCS and AwTV-POCS. This hyper-parameter is involved in the condition which has been discussed earlier when the definitions of hyper-parameters are introduced in section 5.1.1. For the next iteration, the gradient-descent step-size is reduced by α_{red} only when the ratio of change due to TV minimisation to change due to SART is greater than r_{max} and the L_2 error of image in the current iteration is greater than ε simultaneously.

The range of α_{red} being studied here is varied from 0.1 to 1. The RMSE and CC plots of final reconstructed images using different α_{red} values over this range are shown in figure 5-13.



(a) RMSE plots from varying α_{red}



(b) CC plots from varying α_{red}

Figure 5-13: RMSE and CC plots across different α_{red} values

Considering RMSE and CC plots, the reconstructed images obtained from AwTV-POCS with different α_{red} achieve lower RMSE values and higher CC values than that of the ASD-POCS in almost all cases. Again, this is also showing that the adaptive-weighted function can significantly improve the quality of the reconstructed images.

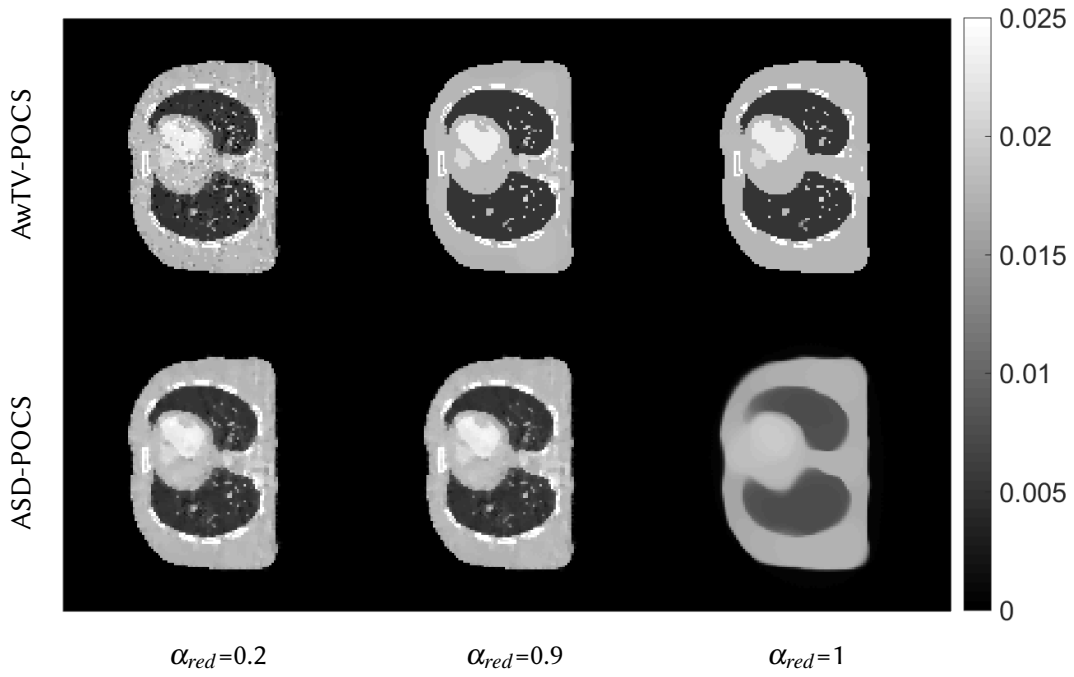


Figure 5-14: Reconstructed images from ASD-POCS and AwTV-POCS with different α_{red} values

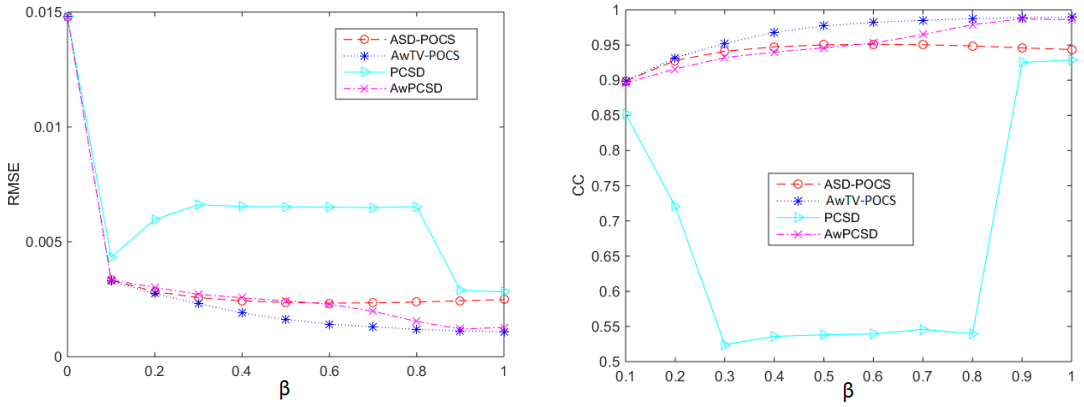
From this study, we can see that the α_{red} hyper-parameter is a sensitive one. A slight increase in value of α_{red} leads to a drastic change in the final reconstructed image. See for an instance the reconstructed images in figure 5-14 when α_{red} is increased from 0.9 to 1. It is a crucial hyper-parameter to set to a proper value. For both algorithms, reduction of α is crucial. Therefore, the α_{red} should not be set to 1 as the quality of image is deteriorated hugely. The suggested value of

α_{red} for the implementation of ASD-POCS and AwTV-POCS is any value close to 1 for the best results.

Since the α_{red} hyper-parameter is sensitive to value change and requires a proper setting, it is another advantage of PCSD and AwPCSD algorithms that these 2 algorithms do not require this hyper-parameter for their implementations.

Relaxation parameter (β)

This is the hyper-parameter that defines how strong the effect of SART function is on the current iteration of the reconstruction. In this study, the relaxation parameter is varied from 0 to 1. According to the RMSE plot in figure 5-15a, when β = 0 , all 4 algorithms have a high level of RMSE which is expected as the SART operation is suppressed. As β increases to 1, the quality of reconstructed images is gradually improved with lower RMSE for all algorithms apart from PCSD where the RMSE values increase in the middle part of the range. Although the RMSE of reconstructed images with β closer to 1 are slightly different, the quality of images is not that significantly improved. Thus, the recommended setting for β hyper-parameter is 1.



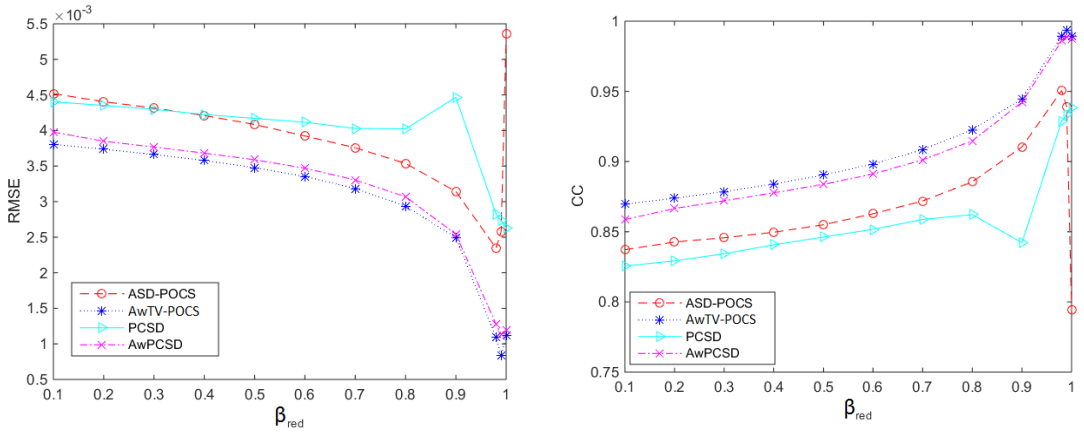
(a) RMSE plots from different β

(b) CC plots from different β

Figure 5-15: RMSE and CC plots across different β values

Reduction factor of relaxation parameter (β_{red})

This hyper-parameter is varied from 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.98, 0.99 to 1. The value of β_{red} = 1 means that the same amount of effect from SART operation is kept constant as the iteration goes on. Decreasing β_{red} from 1 to 0.1 reflects in less and less amount of relaxation parameter β for the next iteration.



(a) RMSE plots from different β_{red} (b) CC plots from different β_{red}

Figure 5-16: RMSE and CC plots across different β_{red} values

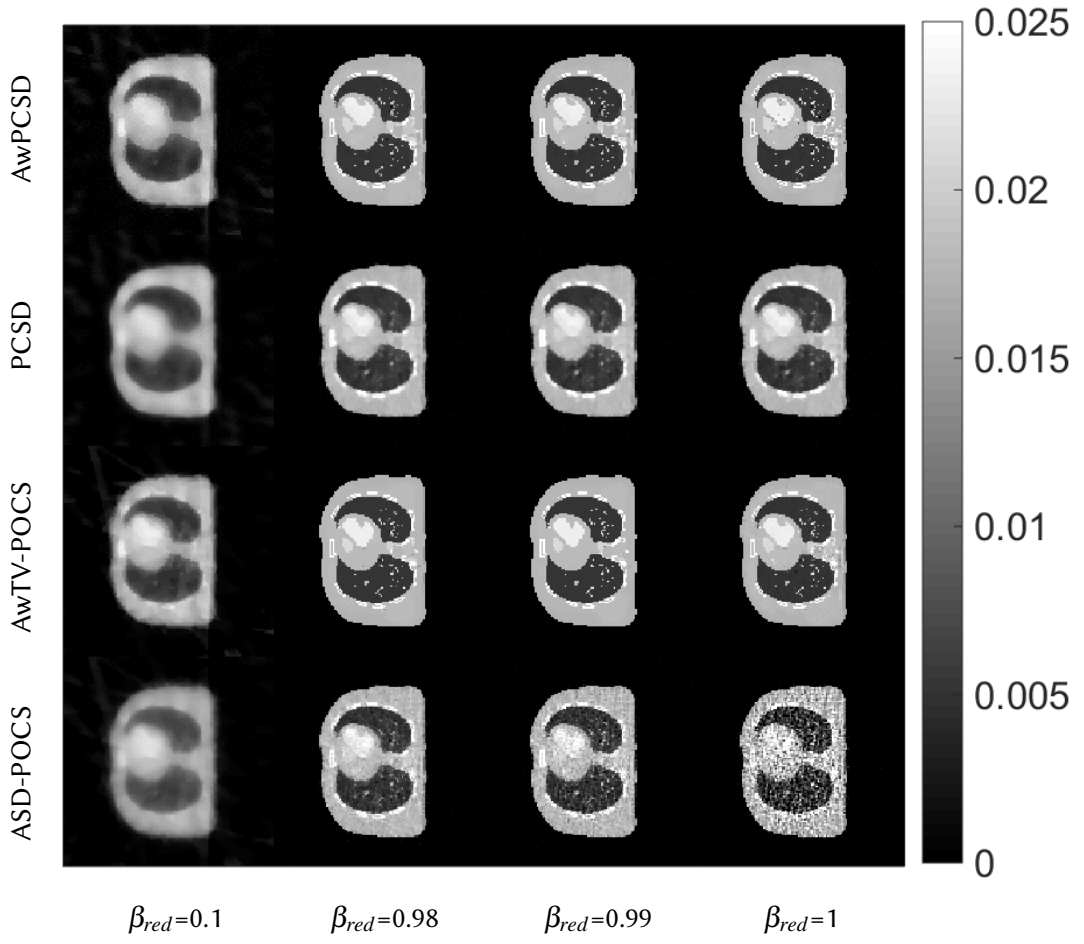


Figure 5-17: Reconstructed images with different β_{red} values

From the RMSE and CC plots in figure 5-16, we can see that values of RMSE are the lowest when β_{red} are close to 1. The quality of reconstructed images quickly deteriorate as the β_{red} get smaller upto 0.1 as can be seen from the cross-sectional slices of the reconstructed images in

figure 5-17.

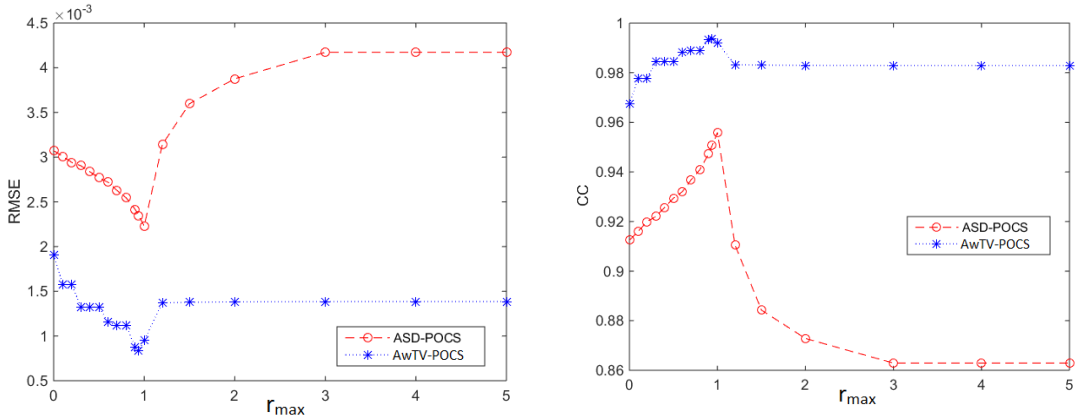
From this study, the suggested range of β_{red} is value larger than 0.98 but smaller than 1 to ensure the effect of SART operation is reduced for the next iteration but not excessive.

Maximum ratio of change by TV minimization to change by SART (r_{max})

The r_{max} hyper-parameter is required by two algorithms: ASD-POCS and AwTV-POCS. This hyper-parameter is also involved in the condition for TV steepest-descent step-size adaptation for next iteration. If the ratio of the change in the image due to steepest descent to the change in the image due to POCS is greater than r_{max} , the gradient-descent step-size is reduced by α_{red}. In case that the current image satisfies the data-inconsistency-tolerance condition, the gradient-descent step-size is then no longer reduced.

From this condition, we can see that the reconstructed images obtained from varying r_{max} also depends on the values of ε and α_{red}. However, the study of combination of hyper-parameters would be too complicated to evaluate for the scope of this study.

The r_{max} in this study is varied from 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.94, 1, 2, 3, 4 to 5. The RMSE and CC plots of the reconstructed images obtained from varying r_{max} in this range is shown in figure 5-18.



(a) RMSE plots from different r_{max}

(b) CC plots from different r_{max}

Figure 5-18: RMSE and CC plots across different r_{max} values

According to figure 5-18, the best results for both algorithms are obtained from r_{max} = 1 which means that the algorithms best perform when the change in the image is balanced between the two operations. In other words, the ratio of the change in the image due to steepest descent should be equal to the change in the image due to SART. The same way as in the previous sections, the introduction of the adaptive-weighted TV norm results in overall a better image.

Scale factor for adaptive-weight TV norm (δ)

This hyper-parameter is only required by AwTV-POCS and AwPCSD algorithms that minimise adaptive-weighted TV norm, as expressed in the equation 4.40 in the previous chapter. The

δ hyper-parameter is a scale factor controlling the amount of smoothing being applied to the image voxel at edges relative to non-edge region during each iteration.

Figure 5-19 shows the weight function plot from the weight equations 4.41-4.43. The range of image gradient in X axis is specified from 0 to 0.01 with six values of δ : 0.0005, 0.001, 0.003, 0.005, 0.01, 0.1. The RMSE and CC values obtained from the reconstructed images of different values of δ are plotted in figure 5-20.

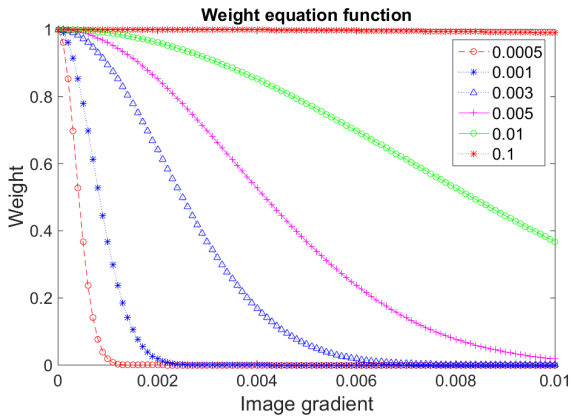
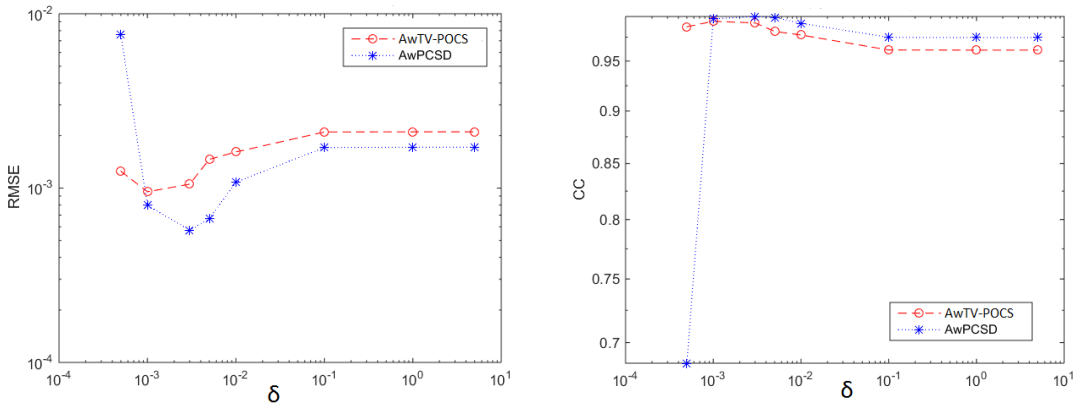


Figure 5-19: Weight equation function



(a) RMSE plots from different δ

(b) CC plots from different δ

Figure 5-20: RMSE and CC plots across different δ values

According to figure 5-19, the weight equation with a small value of δ specifies low weights to almost every pixels. This means that the algorithm preserves most of the gradient by letting TV minimisation have less influence to its implementation . Hence, the reconstructed images will be on the noisy side as can be seen from the figure 5-21, especially the case of AwPCSD with δ = 0.0005.

On the other hand, when δ is large, the function specifies high weights to almost every gradient size of image. This allows TV minimisation to have more influence during the implementation of the algorithm results in the blurred side of the image.

Setting values of δ to these 2 extreme cases makes the algorithm unable to differentiate between the noise which normally contain small gradient and the edges which contain larger

gradient. The appropriate setting of δ is significant to the adaptive-weighted algorithms as it allows TV minimisation to have more influence to remove noise and less influence to preserve edges.

A proper choice of δ is specific to each data set. The suggested way to choose δ is by setting it to approximately 90th percentile of histogram of the reconstructed image from OS-SART algorithm. It should also be noted that this suggestion is derived from the experiment with one set of data, i.e. simulated projection from XCAT phantom. Later on in this chapter, this suggestion of choosing δ will be applied to a different data set, to observe if this suggestion still works.

The weight equation function can be plotted using this value of δ to see how much influence of TV minimisation is preferred for different level of image gradient. Minor alternation might be needed around this value to ensure that the weight equation can preserve the gradient of the edges while removing the potential noise of the image.

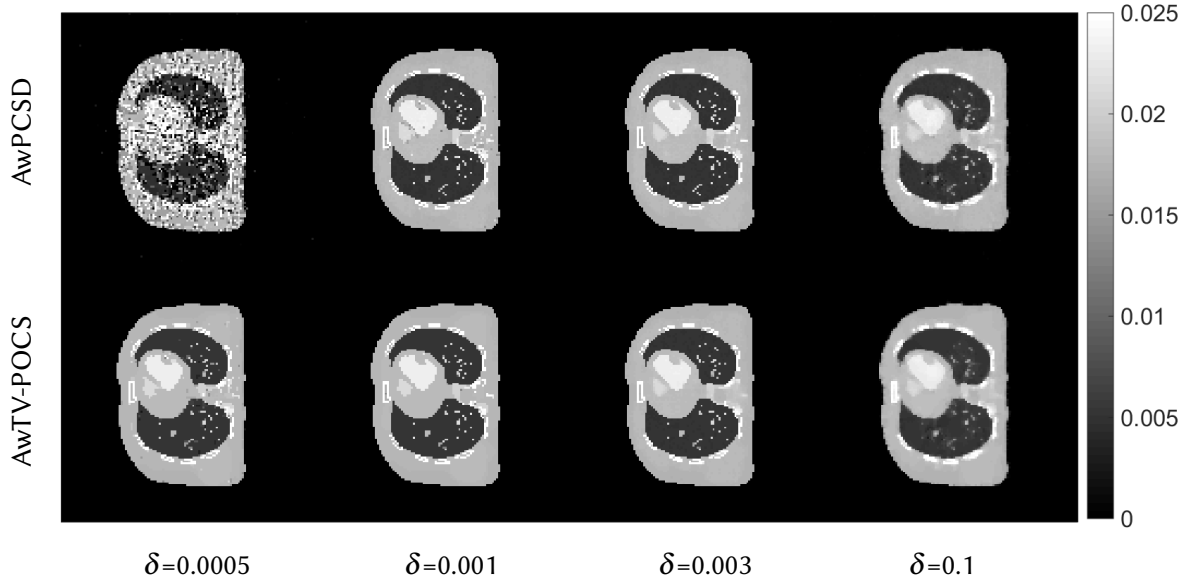


Figure 5-21: Reconstructed images from different δ values

After the sensitivity analysis of all the hyper-parameters has been implemented, figure 5-22 shows the cross-sectional slices of reconstructed images from the best possible setting of hyper-parameters obtained from the analysis. The best values of hyper-parameters from the analysis, as well as the reconstruction times taken for each algorithm are presented in table 5.4. It is clearly seen that the AwPCSD algorithm can preserve the edges of reconstructed image better than other 3 algorithms.

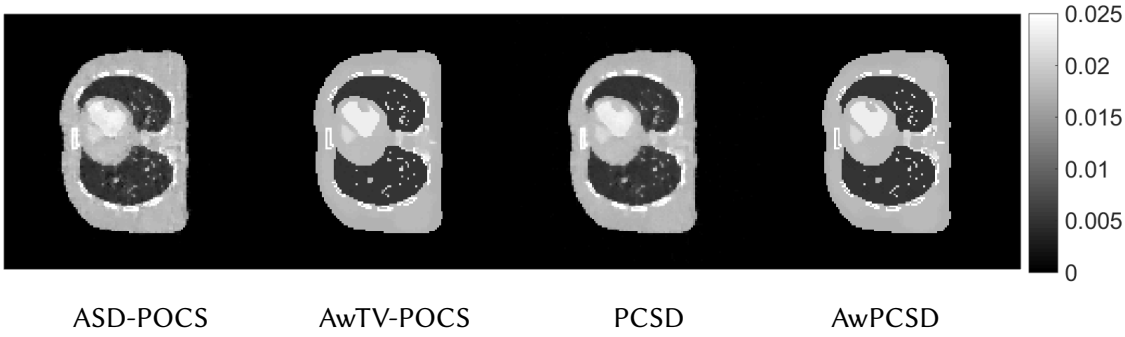


Figure 5-22: The cross-sectional slices of reconstructed images of 4 algorithms from the best set of hyper-parameters.

Table 5.4: The best values of hyper-parameters obtained from the sensitivity analysis and the reconstruction times.

Algorithms	ε	ng	α	α_{red}	β	β_{red}	r_{max}	δ
ASD-POCS $\approx$ 2 mins 17secs	0-40	25	0.002	0.9	1	0.99	1	N/A
AwTV-POCS $\approx$ 6 mins 45secs	40	45	0.002	0.9	1	0.99	1	0.001
PCSD $\approx$ 9 mins 43secs	40	6	N/A	N/A	1	0.99	N/A	N/A
AwPCSD $\approx$ 10 mins 31secs	0-40	6	N/A	N/A	1	0.99	N/A	0.003

5.2.3 Further analysis of the reconstructed images

In order to analyse the edge preserving property of the experimental results, the image profiles along the horizontal and vertical lines as shown in the figure 5-23 are plotted.

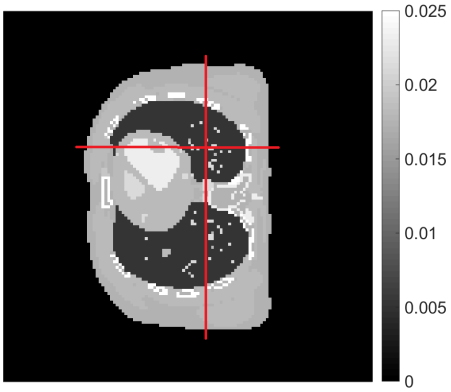
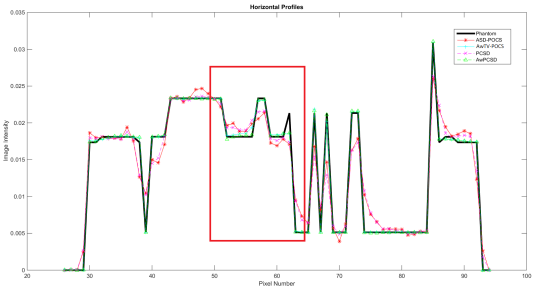
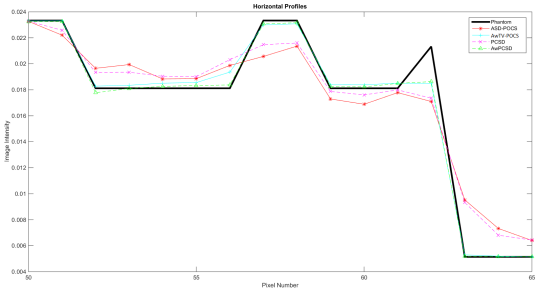


Figure 5-23: The image profiles are plotted along the horizontal and vertical lines.

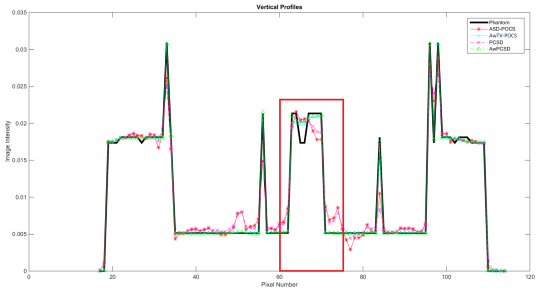
The profiles of reconstructed images from the 4 algorithms are plotted in figure 5-24 with reference to the Thorax phantom. This is to compare the ability of the reconstruction algorithms to reconstruct the small features as well as preserving the edges of the phantom.



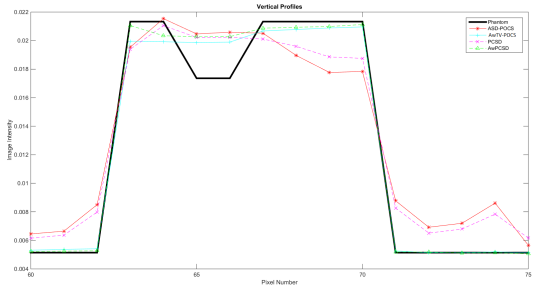
(a) The image profiles along the horizontal line.



(b) Partial profiles of the selected ROI.



(c) The image profiles along the vertical line.



(d) Partial profiles of the selected ROI.

Figure 5-24: 1D profiles along the horizontal line (81st row of the reconstructed images) and the vertical line (71st column of the reconstructed images.)

The 1D profiles of the exact phantom image are plotted in solid black line for reference. Region of interest (ROI) is selected from 1D profiles and marked by the red rectangles in the left column of the figure 5-24. The profiles of these selected pixels are plotted in the right column to observe the differences clearer. It can be obviously seen that the profiles of the adaptive-weighted algorithms i.e. AwTV-POCS and AwPCSD show better alignment with the exact phantom image than those of the ASD-POCS and PCSD algorithms.

Comparing between the two adaptive-weighted algorithms: AwTV-POCS and AwPCSD, although the differences between these two methods are not clearly visible,the reconstructed image from AwPCSD shows a slightly better alignment with the exact phantom image than that of the AwTV-POCS algorithm.

5.3 Experimental evaluation

SophiaBeads dataset

Apart from the simulated data set, the proposed algorithm is also tested with the real microCT datasets, The SophiaBeads Datasets [28]. The sample is a plastic tube with a diameter of 25 mm, filled with uniform Soda-Lime Glass (SiO_2-Na_2O) beads of diameters 2.5 mm (with standard deviation 0.1 mm). The dataset is loaded using the scripts in the project [27]. The source-

to-detector distance is 1.007×10^3 mm and the source-to-object distance is 80.6392 mm. The detector size is 1564 x 1564 pixels and the image size is 1564 x 1564 x 200 voxels. The number of projections used to reconstruct the image in this experiment is 64 projections.

The gold standard image used as a reference in this study is reconstructed by FDK algorithm with 2,048 projections. The proposed algorithm, AwPCSD, is tested with this dataset as well as other 3 TV-based regularization algorithms and FDK for a comparison. The set of hyper-parameters used for each TV-based algorithm is derived as suggested from the experiments of sensitivity analysis in the previous section. The detail of each hyper-parameter is explained in table 5.5

Table 5.5: The optimum set of hyper-parameters used for SophiaBeads dataset

Algorithms	ε	ng	α	α_{red}	β	β_{red}	r_{max}	δ
ASD-POCS	1.5e+04	25	0.002	0.9	1	0.99	1	N/A
AwTV-POCS	1.5e+04	45	0.002	0.9	1	0.99	1	0.1129
PCSD	1.5e+04	6	N/A	N/A	1	0.99	N/A	N/A
AwPCSD	1.5e+04	6	N/A	N/A	1	0.99	N/A	0.0922

The optimum set of hyper-parameters used in this SophiaBeads dataset have some similarities and differences compared to the one used for Thorax phantom dataset. The hyper-parameters such as TV sub-iteration (ng), TV hyperparameter (α), Reduction factor of TV hyperparameter (α_{red}) , Relaxation parameter (β), Reduction factor of relaxation parameter (β_{red}) and Maximum ratio of change by TV minimisation to change by SART (r_{max}) are similar, whereas data-inconsistency-tolerance (ε) and Scale factor for adaptive-weight TV norm (δ) are different. The latter two hyper-parameters are defined specifically for each dataset based on the recommended suggestions in the previous section with some small modification. A cross-sectional slice of the reconstructed images from each algorithm is shown in figure 5-25, together with the RMSE values computed using equation 5.3.

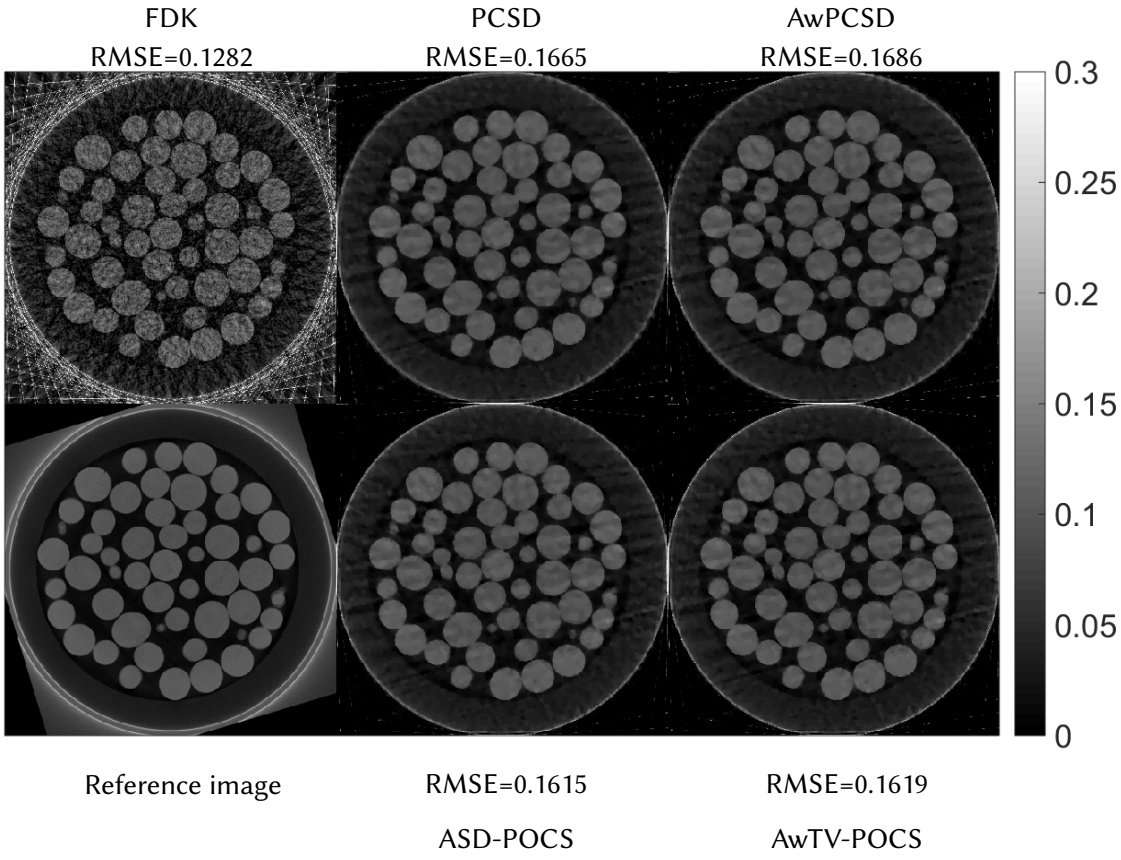
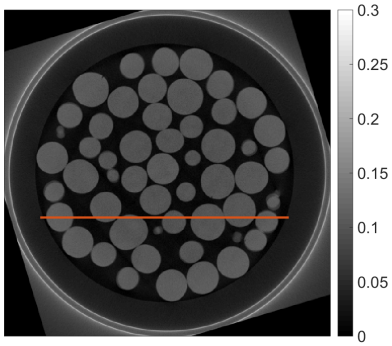
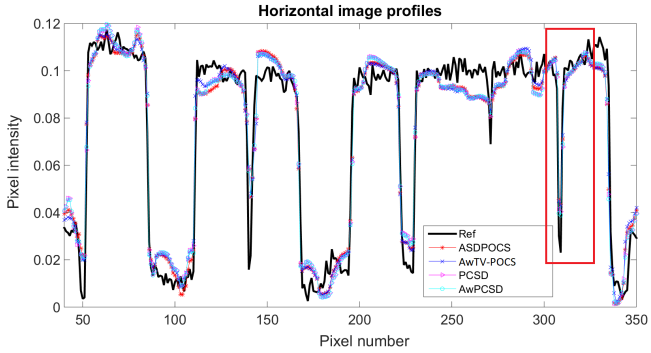


Figure 5-25: The cross-sectional slices of reconstructed images from SophiaBeads datasets.

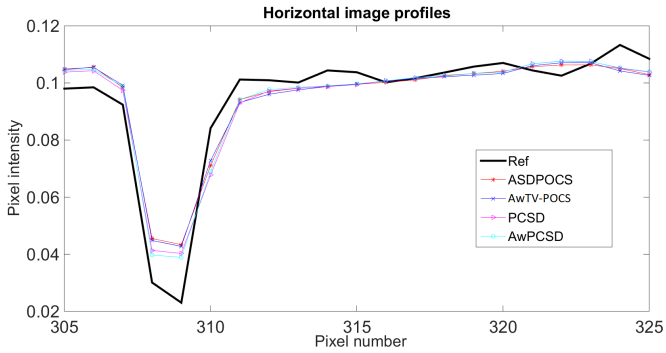
Considering the RMSE values of all the results from 5 algorithms, the lowest value is the reconstructed image from the FDK algorithm. The other reconstructed images from 4 TV-regularised algorithms have higher RMSE values than that of the FDK but rather similar to each ither, with only slightly differences. Although the one of FDK algorithm has the smallest value, the image shows severe artifacts, especially outside the circle of the object. The reconstructed images from TV-regularised algorithms have higher RMSE values but show less artefacts. To observe the difference of each algorithm better, the image profiles along the horizontal line as shown in figure 5-26a are plotted to compare the edge preserving property. The horizontal image profile along the 146th row is plotted in figure 5-26b with the ROI between 305th to 325th column as shown in figure 5-26c.



(a) Horizontal line through the 146th row of the image



(b) The image profiles along the horizontal line



(c) Partial profiles of the selected ROI

Figure 5-26: The reconstructed images profiles

Considering the 1D profile plot of the ROI in figure 5-26c, the reconstructed images from 4 algorithms have very similar image profiles. However, the result from the proposed AwPCSD algorithm is closer to the reference image in some parts, especially when the pixel intensity is lowering down approximately near the pixel number 308-309. This shows that the proposed AwPCSD algorithm performs relatively similar, if not better than the other 3 existing algorithms with less critical hyper-parameters to tune.

5.4 Effects of reconstruction using total variation regularised algorithms from different number of projections

As discussed in the introduction chapter, the frequency of CBCT imaging has increased during the past decades with a rapid growth of IGRT routine due to the superiority of CBCT imaging in volumetric visualisation of patient localisation. As a consequence, there is an increasing awareness of high radiation dose delivered to patients from imaging. This reflects in a number of published papers, which have investigated into the magnitude of imaging dose from CBCT by measurements and calculations, as reported in the review paper [1]. Imaging dose reduction has been an active area of research recently. It is of greater importance for the imaging dose reduction in children than adults, as shown in [34], [33] that the pediatric imaging dose delivered to several organs is approximately twice that of adult. Several methods can be utilised to reduce imaging dose during image acquisition such as adjusting the start/stop angles of the x-ray source to avoid radio-sensitive organs, reducing scan length by using smaller field sizes and using low-dose protocols to image.

The amount of scanning dose is directly related to a number of projections and an exposure per projection, as presented in [146]. The effect of a limited number of projections on the image quality of megavoltage digital tomosynthesis was studied in [112]. The experiments in the work [112] were implemented using portal images, which were acquired using an add-on charge coupled device (CCD) camera EPID. The dose measurements and calculations were implemented to determine the dose delivered to 3 points of the test phantom during image acquisition. The lithium fluoride (LiF) thermoluminescent dosimeters (TLDs) chips were positioned at these points to measure the amount of dose. A total of 41 projection images were acquired every degree from -20° to 20°. Number of projections used to measured the dose in the experiment was from 2 to 41 projections. The results showed the expected linearly increase in dose to all points with an increasing number of projections. Apart from the radiation dose, a number of projections also affects a data acquisition time and reconstruction time. It is, therefore, desired to have a balance between time and quality such that the scan times can be kept as low as possible while still achieving good quality of image.

An interesting question can be raised regarding how many projections are sufficient for an accurate X-ray CT reconstruction. Number of projections directly affects the acquisition time and reconstruction time. Therefore, it is important to have a balance between time and quality so that the scan time can be kept as low as possible while still achieving good quality image from the CT scan. Among existing literature, there is little quantitative guidance regarding how much TV regularisation helps to reduce the number of projections.

In [69], Jørgensen and Sidky developed and adapted phase-diagram analysis from the compressed sensing (CS) for empirical use in X-ray CT, in order to determine how few projections are enough for recovery of an image of a given sparsity. The phase-diagram is a systematic analysis tool from the CS to study and express theoretical relations between sparsity and sufficient sampling. In their work, two theoretical analyses by Donoho and Tanner (DT) [38], [37], [36], [39] and Amelunxen, Lotz, McCoy and Tropp (ALMT) [3] are presented. Brief explanation of each

analysis is given in the following.

For an s-sparse signal $x \in \mathbb{R}^N$ and m samples, the phase-diagram from DT shows recoverability as a function of (ρ, δ), where $\rho = s/m \in [0,1]$ and $\delta = m/N \in [0,1]$. The results in [69] showed that there exist phase-transition curves $\rho(\delta)$ such that at a sampling level of δ all/most signals with $\rho < \rho(\delta)$ can be recovered and all/most signals with $\rho > \rho(\delta)$ failed to be recovered. These are represented as the area below the curve for full recovery and the area above the curve for no recovery. The ALMT phase diagram shows recoverability as a function of $(s/N, m/N) \in [0,1]^2$, where m/N are critical sampling values and s/N are sparsity values. The phase-transition curves of ALMT show that most images of a given sparsity are recovered from more samples than the critical level, and not recovered from fewer samples. The region of full recovery are above the phase-transition curve.

In [69], they had taken the first steps to use the phase-diagrams to predict critical sampling levels for large-scale TV reconstruction by testing with test phantoms from a cone-beam CT scan data set of a walnut. The simulation results reported in their work showed that the critical sampling level was predicted very well by phase-diagram analysis. More investigation regarding how few projections will suffice for accurate TV-regularised reconstruction was done in [68]. Jørgensen et al systematically studied TV-regularised reconstruction quality at reduced numbers of projection using real X-ray CT data. The following 2 questions were raised in their study:

- Does TV-regularised reconstruction compensate better for reduced information from few high-exposure or many low-exposure projections?
- Is there a connection between gradient sparsity and how few projections provide enough information that TV-regularised reconstruction succeeds?

Regarding the first question, it can be concluded from the experimental results shown in [68] that it is more beneficial to distribute low-exposure projections across the highest possible number of projections than few high-quality projections. Regarding the second question, it is inconclusive if there is a possible connection between gradient sparsity and number of projections for the reconstruction in real X-ray CT data set. The experimental results showed no clear difference in their dependence on the number of projections between two data sets with different sparsity levels. However, there is a large error reduction between 64 and 128 projection, which may hint that there is a number of projections below which TV-regularised reconstruction will not be successful.

In order to better understand the effects that different numbers of projections have to the reconstruction quality, an experiment was set up and presented in this section. The XCAT phantom was used again as an input data set to 4 TV-regularised algorithms, i.e. ASD-POCS, AwTV-POCS, PCSD and AwPCSD. The FDK algorithm is also implemented for a comparison purpose. The default Poisson and Gaussian noise were also added to the input projection data to simulate a realistic noise. The numbers of projections used for the reconstruction with 5 algorithms range from 20 projection views to 360 projection views, equally sampled over 360°. After the reconstructed images from all the algorithms with different numbers of projections were obtained,

the root mean squared error (RMSE) of each reconstructed image was calculated by using the equation 5.3. The reference image in this experiment was the exact XCAT phantom image.

The experiments were performed on the computer with an Intel Core i7-4930K CPU@3.40GHz with 32 GB RAM and GPU: NVDIA GeForce GT 610. All the TV regularised algorithms were specified maximum number of iteration of 50. All the hyper-parameters for each algorithm were specified as suggested from the results of sensitivity analysis presented earlier in this chapter. The $ng, \alpha, \alpha_{red}, \beta, \beta_{red}$ and r_{max} for each algorithm was specified according to the table 5.4. The ε hyper-parameter was specified as the L_2 norm of the reconstructed image obtained from OS-SART algorithm. The δ hyper-parameter was specified as approximately $90th$ percentile of histogram of the reconstructed image from OS-SART algorithm. Values of ε and δ for different numbers of projections are shown in table 5.6, together with the reconstruction times per iteration and iteration numbers when the 4 TV regularised algorithms stopped.

Table 5.6: Variation of different ε and δ values, reconstruction times per iteration, the iteration numbers where the TV regularised algorithms stopped with different numbers of projections.

Numbers of projections	**Values of ε / δ (for Aw algos)**	**ASD-POCS / stop at**	**AwTV-POCS/ stop at**	**PCSD / stop at**	**AwPCSD / stop at**
20	107.2733/0.0201	1 s/iter 17	1.3 s/iter 14	1.1 s/iter 50	1.2 s/iter 50
40	101.5651/0.0209	1.8 s/iter 13	2.4 s/iter 12	2.1 s/iter 50	2.4 s/iter 50
60	114.5665/0.0213	2.7 s/iter 13	3.5 s/iter 10	3.1 s/iter 50	3.5 s/iter 50
80	124.9976/0.0216	3.6 s/iter 10	4.7 s/iter 8	4 s/iter 50	4.6 s/iter 50
100	135.4417/0.0218	4.5 s/iter 9	5.8 s/iter 8	4.9 s/iter 50	5.8 s/iter 50
120	145.8427/0.0218	5.2 s/iter 8	6.9 s/iter 7	5.9 s/iter 50	6.9 s/iter 50
140	153.8287/0.0220	6.2 s/iter 8	8.2 s/iter 6	6.5 s/iter 50	7.6 s/iter 50
160	163.3692/0.0220	7 s/iter 7	9.2 s/iter 6	7.9 s/iter 50	9.1 s/iter 50
180	171.0661/0.0221	7.7 s/iter 7	12.5 s/iter 6	8.7 s/iter 50	9.6 s/iter 50
200	178.6926/0.0222	8.7 s/iter 7	12.8 s/iter 6	9.7 s/iter 50	10.8 s/iter 50
220	187.5277/0.0223	9.5 s/iter 6	13.1 s/iter 13	10.9 s/iter 50	11.9 s/iter 50
240	194.1640/0.0224	10.1 s/iter 6	14.1 s/iter 19	11.8 s/iter 50	13.9 s/iter 50
260	200.3112/0.0224	11.5 s/iter 6	15.2 s/iter 13	12.7 s/iter 50	14.1 s/iter 50
280	207.5493/0.0225	11.9 s/iter 8	16.2 s/iter 31	13.8 s/iter 50	15.1 s/iter 50
300	214.3009/0.0225	12.8 s/iter 14	17.8 s/iter 20	14.7 s/iter 50	16.2 s/iter 50
320	221.0016/0.0226	13.7 s/iter 16	19 s/iter 21	14.5 s/iter 50	17 s/iter 50
340	228.5531/0.0226	14.2 s/iter 21	20.2 s/iter 44	15.3 s/iter 50	18.1 s/iter 50
360	234.1947/0.0227	15.2 s/iter 50	20.1 s/iter 44	17.3 s/iter 50	20.4 s/iter 50

In this experiment, the ASD-POCS and AwTV-POCS algorithms stopped by satisfying the stopping criterion of cosine angle between the TV and data-constraint gradients and the L_2 norm error between the measured projections and the projections computed form the estim-

ated image in the current iteration, as explained in equation 5.1. With different numbers of projections, these 2 algorithms stopped at different number of iteration, as presented in table 5.6. The PCSD and AwPCSD algorithms were implemented until the maximum number of iteration of 50 was reached. We can also observe from the table 5.6 that the reconstruction times of all the algorithms increase with an increasing number of projections used for the reconstruction. Furthermore, it can be observed that most of the ε and δ values slightly increase as the number of projections increases in each step.

The RMSE values of the reconstructed images from 4 TV regularised and FDK reconstruction algorithms with different numbers of projections from 20 projections to 360 projections are shown in figure 5-27. The Y axis of the graphs is plotted in a log scale.

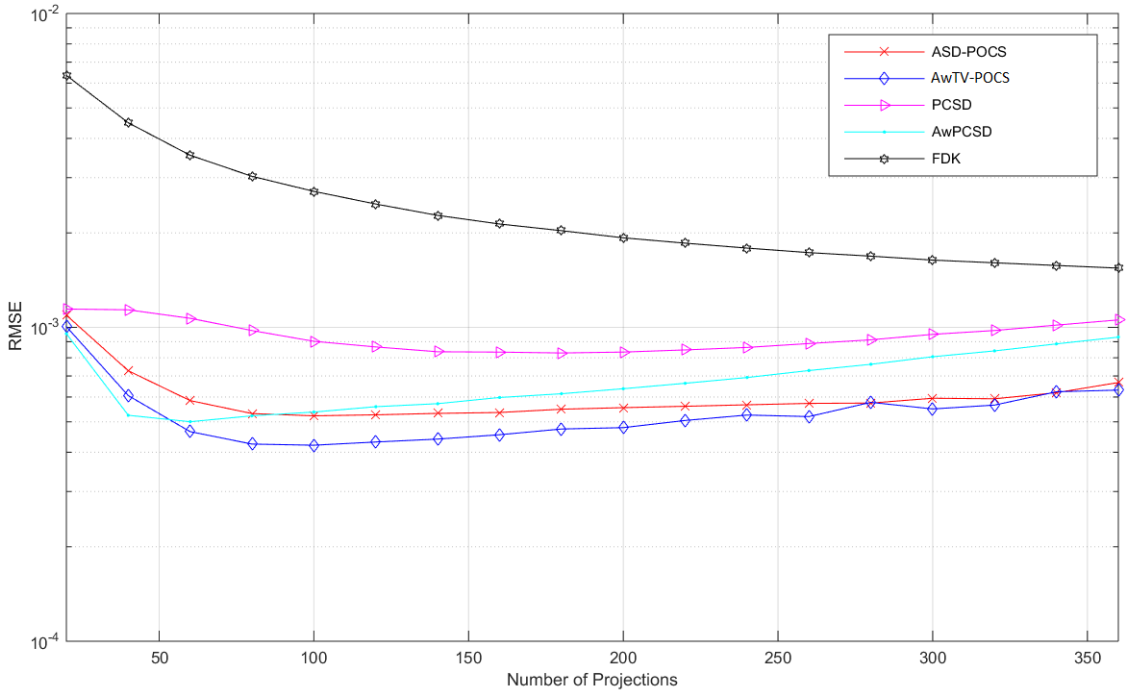


Figure 5-27: RMSE values of the reconstructed images from ASD-POCS, AwTV-POCS, PCSD, AwPCSD and FDK algorithms with different numbers of projections used for reconstructions, i.e. varying from 20 projection views to 360 projection views, equally sampled over 360°. The Y axis of the graphs is plotted in a log scale.

Figure 5-28 depicts some examples of the reconstructed images from this experiment. The reconstructed images shown in figure 5-28 are the reconstruction results of 4 TV regularised algorithms and FDK algorithm by using 20,60,200 and 360 projection views.

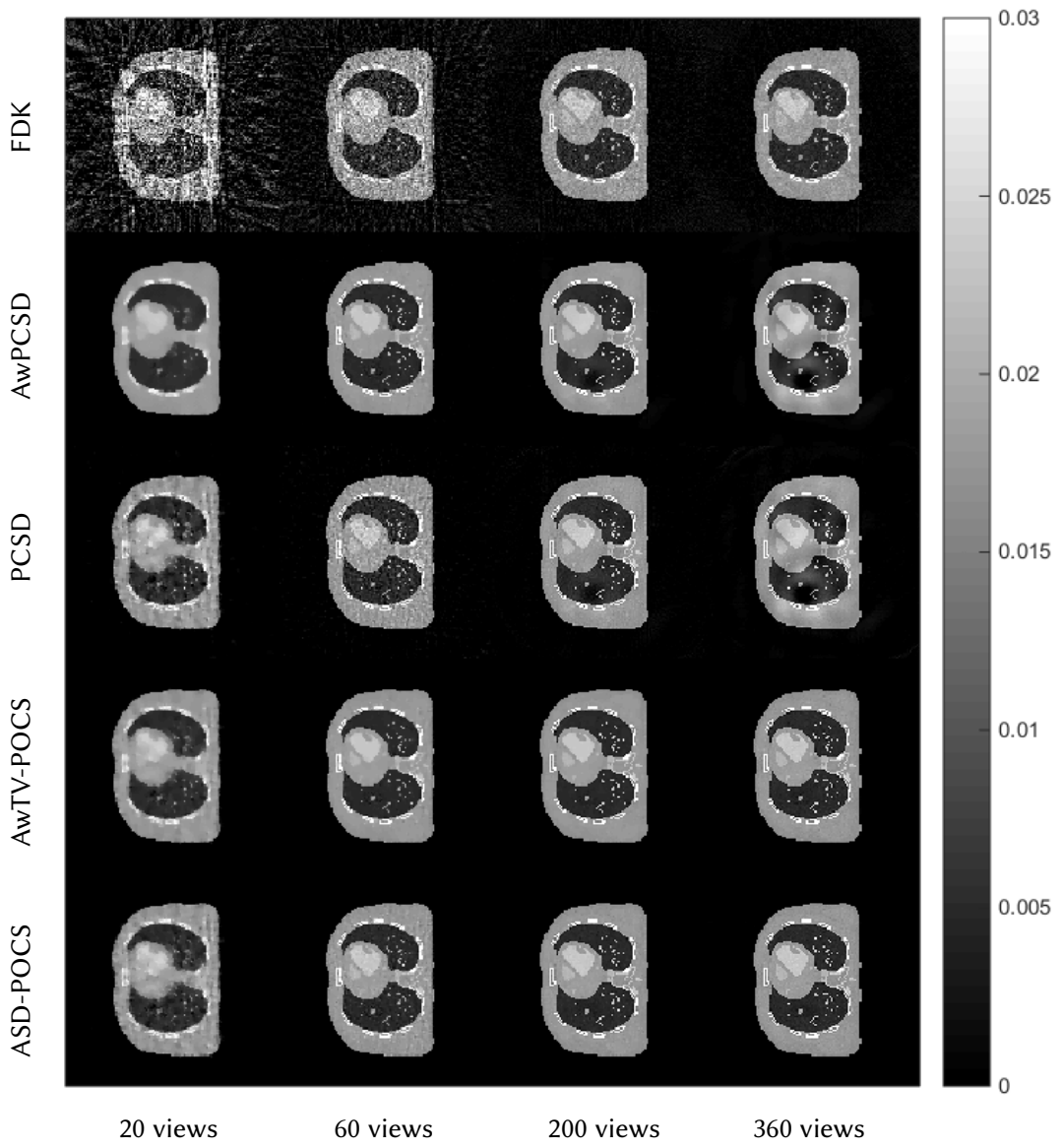


Figure 5-28: Reconstructed images from ASD-POCS, AwTV-POCS, PCSD, AwPCSD and FDK algorithms obtained using 20, 60, 200 and 360 projection images.

According to the RMSE plots on figure 5-27, it can be observed that, as the numbers of projections increase from 20 to 360, only the RMSE values of reconstructed images from FDK algorithm constantly decrease. These results of FDK algorithm are in line with the reconstructed images shown in the top row of figure 5-28. As the number of projections increases from 20 to 360 projections, the FDK reconstructed images becomes less noisy with an improved quality of images. Next, we consider the results of the TV regularised algorithms. The RMSE values of 4 TV regularised algorithms start off decreasing and then eventually increase, especially when number of projections are in a high region, ≈ 300 and more. Among all the 4 algorithms, the PCSD algorithm exhibits the slowest decreasing of RMSE values at the beginning and the increasing at the end. Frankly speaking, these behaviours of the 4 TV regularised algorithms are not in line with an assumption that the reconstructions using higher numbers of projections should result

in lower RMSE values. However, visual inspection of figure 5-28 reveals that the reconstructed images from these 4 algorithms become less and less noisy with clearer small features recovered with an increasing number of projections. Some prominent dark and white circles artifacts can be seen in the lower part of reconstructed images from PCSD and AwPCSD algorithms from 200 views, even become more obvious in 360 views. These artifacts may be the reason of high RMSE values of the reconstructed images from PCSD and AwPCSD algorithms towards the end of the graphs in figure 5-27.

The conclusions can be drawn from this experiment in several aspects. First of all, the TV regularised algorithms perform better than the FDK algorithm when the number of projection is limited, especially in the 20 views case. This can be seen from the lower RMSE values and less noisy reconstructed images. Secondly, this experiment reinforces the point that finding an appropriate set of hyper-parameters for the TV regularised algorithms is a difficult task. This is because there are many factors involved in choosing hyper-parameters such as a data set, sizes of the measurement data and image to be reconstructed, as well as a number of projections used for a reconstruction. Different numbers of projections require different values of hyper-parameters for the algorithms to work properly and produce a good quality image. For example, more TV sub-iteration number might be required for the reconstruction with low number of projections, in order to deal with noise of a reconstruction with limited data.

5.5 Summary

In this chapter, the hyper-parameter selection of TV-based regularization algorithms is investigated. The sensitivity that the reconstruction image has to value change on each hyper-parameter is analysed, in order to know which ones to prioritise when tuning the algorithms to minimise or completely avoid rerunning the reconstruction with different values of hyper-parameters.

In addition, the robustness of the newly proposed AwPCSD algorithm is tested in comparison with other 3 algorithms: ASD-POCS, AwTV-POCS and PCSD. The sensitivity analysis is evaluated experimentally by two image quality metrics: Root mean squared error (RMSE) and Correlation coefficient (CC). The edge preserving property of the adaptive-weighted function is also analysed using the one-dimensional profiles plot along the horizontal and vertical lines of the reconstructed images from the TV-based algorithms in comparison to the reference image.

The suggested ways of selecting the values for each hyper-parameter are presented in detail in the results section. It is clearly seen from the results that hyper-parameter choice is crucial for the implementation of TV-based regularization algorithms, especially for the following three hyper-parameters: TV hyperparameter (α), Reduction factor of TV hyperparameter (α_{red}) and Maximum ratio of change by TV minimization to change by SART (r_{max}). These hyper-parameters are the most sensitive ones and require careful selection of values. Setting these hyper-parameters to certain values can significantly deteriorate the quality of final reconstructed image.

With this knowledge, it is a great advantage of PCSD algorithm as well as the proposed AwPCSD algorithm because they do not require the mentioned hyper-parameters, making them

a lot easier to implement and less prone to errors compared to the ASD-POCS algorithm.

However, the performance of PCSD algorithm is unreliable at times in response to changes of some hyper-parameters such as data-inconsistency-tolerance (ε), TV sub-iteration number (ng) and Relaxation parameter (β). In these scenarios, the proposed AwPCSD algorithm shows significant robustness over PCSD algorithm by preserving edges of the reconstructed image better.

The minimisation of adaptive-weighted TV norm shows great performance in preserving the edges of the reconstructed algorithms for both two adaptive-weighted algorithms: AwTV-POCS and AwPCSD. This edge-preserving function make the adaptive-weighted algorithms a lot more robust when compared to other two non-adaptive-weighted algorithms, especially for the pair of PCSD and AwPCSD algorithms.

There are some limitations of the sensitivity analysis of combination of hyper-parameters. For some hyper-parameters such as the reduction factor of TV hyperparameter (α_{red}), the gradient-descent step-size for the next iteration will only be reduced by α_{red} when the ratio of change due to TV minimisation to change due to SART is greater than r_{max} and the L_2 error of image in the current iteration is greater than ε simultaneously. This means that all three hyper-parameters can affect the results of sensitivity analysis of α_{red}. However, the sensitivity analysis is done by varying values of one hyper-parameter at a time as the study of combination of hyper-parameters would be complicated to evaluate and is beyond the scope of this study.

The proposed AwPCSD algorithm has shown significant robustness compared to other three existing algorithms: ASD-POCS, AwTV-POCS and PCSD. This AwPCSD algorithm is able to preserve the edges of the reconstructed images better with less sensitive hyper-parameters to tune. Towards the end of the chapter, the effects of reconstruction using TV regularised algorithms and the FDK algorithm with different numbers of projections was studied.

Chapter 6

Hyper-parameter selection using Ant Colony Optimisation algorithm

It has been clearly shown from the results in the previous chapter that the selection of hyper-parameters is of the utmost importance for the implementation of TV-based reconstruction algorithms. If the hyper-parameters are not specified the proper values, the efficacy of the reconstruction algorithms might not be at their best. The proper values of TV hyper-parameters are the ones that are most suited to a set of data and imaging application, for which they make the TV regularised algorithms produce images with good quality for certain tasks. The interactions among a set of hyper-parameters are also complex. This makes the knowledge of picking a proper value for a given task scenario only be acquired through many trials-and-errors, which is a tedious and time-consuming process. The manual adjusting of the hyper-parameters is commonly seen in a number of researches in this field [122], [65], [64], [66], [23]. The complication of hyper-parameter selection makes it difficult to unlock the true potential of iterative reconstruction algorithms in clinical use [148].

In the literature, many techniques are proposed to automatically tune the hyper-parameters in different applications. In [51], Golub *et al* proposed the generalised cross-validation method for estimating the ridge parameter in ridge regression. P.C. Hansen [54] proposed methods to choose the regularisation parameters by locating a characteristic L-shaped corner of a plot between a seminorm of a regularised solution versus a norm of the corresponding residual error. In [108], the method to select regularisation parameter for nonlinear iterative image restoration and MRI reconstruction was proposed. In their work, they demonstrated that minimising predicted-SURE (Stein's unbiased risk estimate) consistently lead to reconstruction with almost optimal value of mean-squared-error (MSE). Also, minimising generalised cross-validation (GCV) yields result with near-MSE-optimal for image restoration and slightly sub-optimal for MRI.

For x-ray CT reconstruction, there has also been an on-going effort to address the problem of hyper-parameter selection in iterative algorithms. In 2009, Xu and Mueller proposed the data-driven learning approach to automate the parameter selection for CT reconstruction tasks by matching given data configurations with their most effective reconstruction parameter con-

figurations [136]. The ordered subsets SIRT (OS-SIRT) is used to demonstrate the principles of their proposed approach. Later on, they devised a parameter space navigation interface that allows users to interactively assist parameter selection for iterative CT reconstruction algorithm (OS-SIRT) [137].

In [138], Xu and Mueller proposed two approaches to learn effective parameter settings for GPU-accelerated iterative CT reconstruction algorithms. The first approach starts by producing a set of reconstructions from the projection data. These reconstructions are evaluated fitness scores by using a quality metric. Adaptive sampling are used to drive the data collection into parameter regions that produce more diverse reconstruction results in terms of the quality metrics. All the reconstructions is then labelled either 'quality, given a certain wall-clock time limit' or 'reconstruction speed, given a certain quality threshold'. The observations with the higher marks, subsequently receive higher weights in determining the reconstruction algorithm parameters.

The second approach of the work proposed by Xu and Mueller is to solve the parameter learning problem as a multi-objective optimisation (MOO) problem. The genetic algorithms (GA) are well suited for this kind of problem. More detail about the GA will be explained in the next section. In their work, the nondominated sorting genetic algorithm (NSGA) is used to find a nondominated optimal solution set. The parameter learning process in this approach is executed by firstly generating the initial population of parameter. The OS-SIRT reconstruction algorithm and the parameter searching algorithm NSGA-II are running alternatively until the stopping criterion has been satisfied. Within each generation, OS-SIRT is implemented as a parallel fitness values evaluator, while the NSGA-II combines these fitness values to find the current solution set with a user-defined set size. The results showed that the second approach using GA can provide optimal parameter settings considerably faster than the first approach because fewer reconstructions need to be computed. Their work shows the feasibility of using GA as a tool for parameter selection in iterative algorithms and reveals interesting relationships among the parameters tested. However, their proposed approach is not scalable in the number of parameters and uses the OS-SIRT reconstruction algorithm in conjunction with bilateral filtering for regularisation.

The scalable framework of the learning of effective parameter setting is proposed in [148] by Zheng et al. In their work, a compute-based assistant called dose, quality and speed (DQS) advisor is devised. It is a visual interface system that allows users to balance the three most important CT metric; dose, quality and speed. Their optimisation system is designed by using OS-SIRT algorithm with 2D NLM filter and unsharp masking for regularisation. The most effective parameter settings for a set of DQS configurations are generated by ant-colony optimisation (ACO) algorithm, which is a specific form of GA. A visual interface presents the outcome of the optimisation, while a matrix display allows users to compare the corresponding images. The ACO approach used for parameter searching in the DQS advisor is based on their automatic parameter optimisation framework proposed in [147], which will be presented in more detail in section 6.2.

6.1 Hyper-parameter selection framework

First of all, this section explains the framework of the hyper-parameter selection problem to help the reader understands the context, as well as the practical constraints of the problem. As parts of the work done in this thesis, 2 hyper-parameter selection algorithms are developed to demonstrate the possible ways that the computer can be used to select the best set of hyper-parameters from the ranges of user-defined values. The aim is to alleviate the difficulty of choosing the values of hyper-parameters for the implementation of TV regularised algorithm, which can be problematic for the users who are not experts in the field. Upon achieving this, the following points are important factors to be considered:

- **Image reconstruction hyper-parameters**
  Depending on which reconstruction algorithm is being implemented, different algorithms require different numbers of hyper-parameters for their implementations. For instance, the fundamental iterative algorithms such as ART, SIRT, SART and OS-SART require mainly 2 hyper-parameters; relaxation parameter and number of iteration. The CGLS algorithm requires one hyper-parameter, which is the number of iteration. For TV regularised algorithms, as presented in detail in the chapter 5, the ASD-POCS algorithm requires 7 hyper-parameters: $\varepsilon, ng, \alpha, \alpha_{red}, \beta, \beta_{red}, r_{max}$, the AwTV-POCS algorithm requires the same amount of hyper-parameters plus one extra: δ. The PCSD algorithm requires 4 hyper-parameters: $\varepsilon, ng, \beta, \beta_{red}$ and the AwPCSD requires the same amount as that of the PCSD algorithm plus δ.

- **Practical constraints**
  There are several constraints that need to be considered for the hyper-parameter selection problem. The first one is the time constraint, which is directly related to the maximum number of iterations specified for the algorithms. The algorithms might be stopped from meeting the convergence criteria. Otherwise, they will continue implementing until the maximum number of iterations is reached. Another constraint is the image quality, for which the level of image quality might depend on the applications that the images will be used in. This is another input required from the user.

- **User-defined ranges of hyper-parameters** The hyper-parameter selection problem for the reconstruction algorithms is still an unsolved problem. Typically, the set of hyper-parameters which produces good quality of image is obtained from the opinion of experts in the field. Otherwise, it is obtained through trials-and-errors of manual tuning, which is a tedious and time-consuming process. In the context of this work, the hyper-parameter selection algorithms work based on the initial set of hyper-parameters as defined by the general users. The ranges are based on the experimental results as presented in the chapter 5. However, any discrete values of the hyper-parameters can be specified for the hyper-parameter selection algorithms.

Based on the above mentioned factors, 2 hyper-parameter selection algorithms are developed

to demonstrate the possible ways of solving the hyper-parameter selection problem for the TV regularised algorithms. The aim is for the general users who do not have much experience in the field. As an example, the AwPCSD algorithm is chosen as a base reconstruction algorithm, to be combined with the ACO and the Hedge algorithm to choose the hyper-parameters. However, the algorithms are generalised to be applied with any reconstruction algorithms at all. In this chapter, the hyper-parameter selection using the ACO algorithm is explained and the Hedge algorithm will be explained in the next chapter.

6.2 Hyper-parameter selection using Ant Colony Optimisation (ACO)

The ACO approach imitates the way the ants in nature find the shortest path from their nest to the food source. Intuitively, an ant chooses the path randomly from multiple paths available, which have different distances to the food source. Then, it travels back to the nest and leaves pheromones along the trail it has traversed so the other ants can follow this path. As other ants follow the pheromone trails, they will also emit pheromones to reinforce the trails. After some time, the pheromone will evaporate. However, the pheromones on the shortest path will have the least amount of time to evaporate. Eventually, the shortest path will have the highest amount of pheromone deposited, which attract more ants to follow this path. At the end, all ants will converge to the shortest path.

The Ant Colony Optimisation (ACO) is a swarm intelligence approach to search for good paths in discrete graphs, as proposed in [40]. It is a modification of the Ant System algorithm, which was first introduced by Dorigo et al in [41], [42]. The ACO approach is a probabilistic technique and has been successfully implemented in a numerous number of optimisation problems as reported in [40]. It belongs to the class of metaheuristic approaches, which can provide approximate solutions when the global optimum is unobtainable due to incomplete information. In [40] the ant colony optimisation (ACO) metaheuristic is proposed. Its use in hyper-parameter optimisation in CT reconstruction context is not extensive. As briefly mentioned earlier, Zheng et al [148] employed this approach to learn the best parameter setting in their iterative algorithm.

In this chapter, we use the ACO approach for the complex non-linear problem of TV hyper-parameter selection. This work uses the AwPCSD algorithm as TV regularised reconstruction method, but any other TV regularising algorithm can be tuned using the exact same approach. The reason for choosing this algorithm is that the AwPCSD implements the adaptive-weighted TV regularisation, which is able to preserve the edges of the reconstructed image better with less sensitive hyper-parameters required.

6.2.1 The method

To explain how the algorithm works in a nutshell: There is a colony of artificial ants that swarm through the AwPCSD algorithm. Each ant chooses a set of hyper-parameters required for its iterative CT reconstruction. After the reconstruction process finishes, the reconstructed image

obtained from each set of hyper-parameters that belongs to each ant is given a score, which measured using a correlation coefficient (CC). According to this score, each ant leaves a pheromone through the path it has chosen. A higher score means stronger pheromone to attract more ants in the next generations. The reconstructed image with the best score of one generation of ants is kept and used as a base image for the next iteration of algorithm. This approach iterates until either the stopping criterion as explained in equations 5.1 and 5.2 are met, or the maximum number of iteration is reached.

The overall picture of the hyper-parameter selection algorithm using ACO method for CBCT reconstruction is shown in figure 6-1. The detailed explanation of the method in this section is based on this diagram.

Starting with the initialisation stage, the initial images are defined. These are previous iteration best image (*prev-iter-best*) and previous generation best image (*prev-gen-best*), which are both defined to be zero images as a starting point. Also, the pheromones of all values of hyper-parameters are initially set to 1, such that a probability for an individual ant to choose any choices is equal.

In the context of problem in this work, the number of hyper-parameters to be selected is five as required by the implementation of the AwPCSD algorithm. The recommended settings of some hyper-parameters as found from previous chapter are followed, i.e. relaxation parameter (β) is set to 1, reduction factor of relaxation parameter (β_{red}) is set to 0.99, scale factor for adaptive-weighted TV norm (δ) is set to the 90th percentile of the histogram of the reconstructed image from the OS-SART algorithm. Apart from these, there are still two hyper-parameters , data-inconsistency-tolerance parameter (ε) and TV sub-iteration number (ng), that require a proper setting. In the proposed algorithm, the ACO algorithm is employed to identify the optimal hyper-parameter setting for the AwPCSD reconstruction algorithm for a given set of limited projection data.

In each iteration of the algorithm, the process loops through each ant in the colony. The *prev-iter-best* is used as a base image for all ants. An individual ant chooses the value of each hyper-parameter setting following the probability. Ants are not allowed to choose a set of hyper-parameters that was already chosen by another ant of the same generation. The pheromone is computed using the following equation:

$$P_i = \frac{\tau_i}{\sum_{q=0}^{R} \tau_q} \tag{6.1}$$

where τ_i is the pheromone for the choice of hyper-parameter value i and R is the total number of available choices for the hyper-parameter. In the first iteration, the pheromones of all the hyper-parameters are set to 1, making it possible for each ant in the first generation to freely choose any choices.

Each ant then moves on to the reconstruction process, carrying choices of hyper-parameters it has chosen. This process involves running an instance of the AwPCSD reconstruction algorithm.For each run of the AwPCSD algorithm, one iteration of the algorithm is performed. The reconstructed image from each ant is compared with the reference image and a correlation

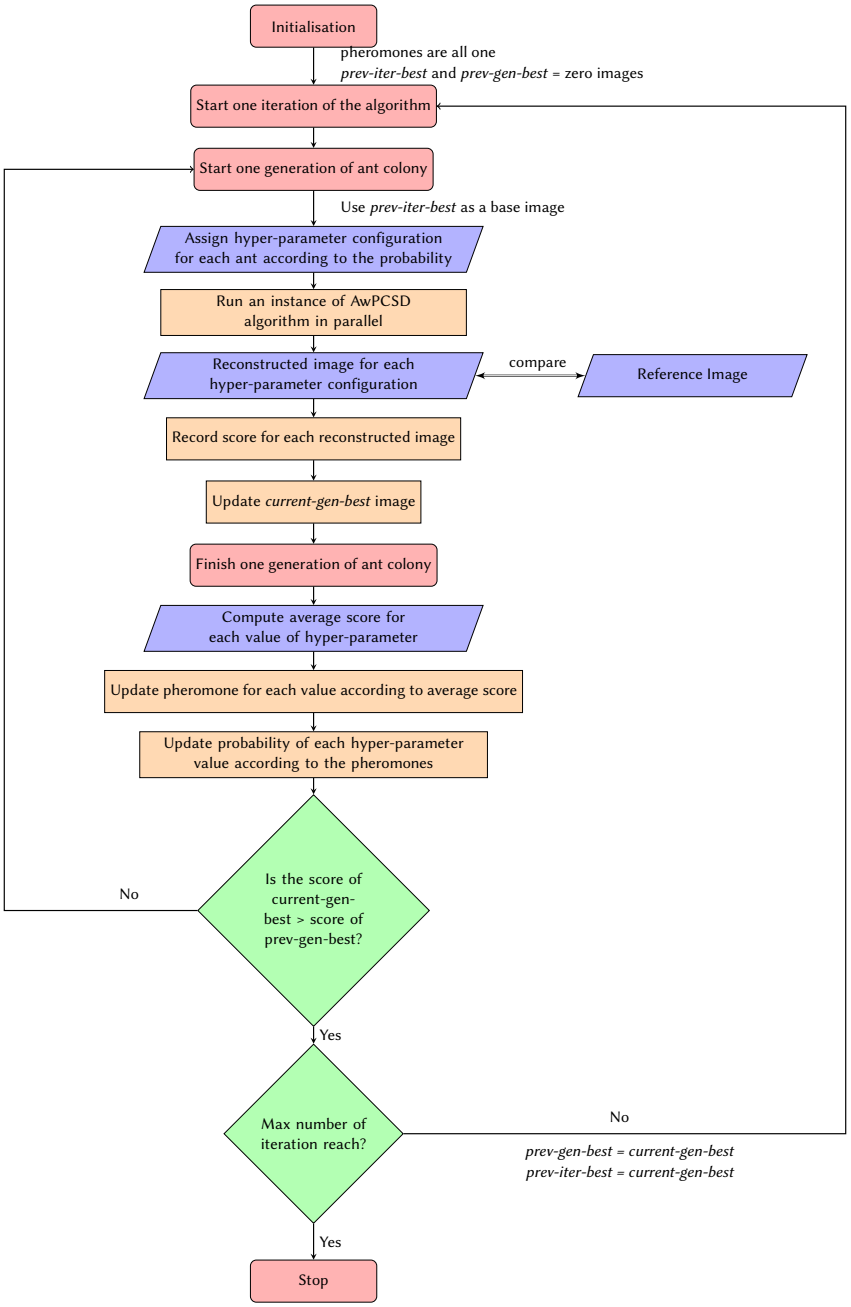


Figure 6-1: The overall picture of the implementation of the computer-aided hyper-parameter tuning approach works

coefficient (CC) is computed and used as a score. This step is repeated until all ants in a colony has finished their moves. The reconstructed image with the highest score among all the ants in that generation is kept as the *current-gen-best* result.

After one generation of ant is finished, the score of each reconstructed image obtained from the reconstruction using each hyper-parameter configuration is separately recorded for each hyper-parameter value in that configuration. Then, all the scores of each hyper-parameter value in that generation of ant is averaged. This average score is used to update the pheromone for that choice of hyper-parameter value for the next generation of ant colony. The pheromone update equation is:

$$\tau_i^{m+1} = (1-\sigma)\tau_i^m + \bar{s}_i \tag{6.2}$$

where s_i is the normalised score of an ant choosing the choice i of hyper-parameters, $\bar{s}_i$ is the average score of all ants choosing the choice i of hyper-parameters. In this study, the CC metric is used as a score. τ_i^m is the pheromone of the choice i of hyper-parameters for the m^{th} generation of ants, σ is the pheromone evaporation factor with the value between 0 and 1. The values of pheromone are normalised between 0 and 1. The pheromone for each choice of hyper-parameter is updated based on the score obtained from the reconstructed image. This is to ensure that the choice which produces high-score reconstructed image attracts more ants in the next generation of ant colony.

Then, two conditions are checked. The first condition is checked whether the score of the *current-gen-best* image is greater than that of the *prev-gen-best* image. If the answer is no, the algorithm launches another generation of ant colony. If the answer is yes and the maximum number of iteration is not reached yet, the algorithm launches the new iteration and replacing *prev-iter-best* and *prev-gen-best* images with *current-gen-best* image. Apart from these 2 conditions, the stopping criterion as explained in the equations 5.1 and 5.2 of the chapter 5 are also checked for each reconstructed image. If these stopping criterion are met, the algorithm is stopped. Otherwise, the whole process is repeated until the user-defined maximum number of iteration is reached. At the end of the implementation, the hyper-parameter configuration with the highest score is an optimal hyper-parameter setting for a given data. Furthermore, the reconstructed image from this setting is obtained as an optimal result from the AwPCSD reconstruction algorithm.

The ACO approach in the proposed algorithm is able to identify optimal hyper-parameters appropriate for a dataset and produce good quality reconstructed image using limited numbers of projection data. The whole process does not require any knowledge about the iterative reconstruction algorithm from the user, nor the intervention between its implementation. The reconstructed image from the proposed algorithm is compared with the results from other ways of choosing parameters, visually and quantitatively. In addition, the optimal set of hyper-parameters from the proposed algorithm is used to reconstruct images from the projection data with different levels of noise and different angle arrangements. The same set of experiments were also tested with different imaging samples to demonstrate the robustness of the proposed

algorithm.

6.3 Experimental results

6.3.1 Image quality metric

The image quality metric which is used to compute a score for the hyper-parameter selection algorithm is the correlation coefficient (CC). This metric measures the degree to which the two images are associated. The CC metric is defined in the following equation.

$$CC = \frac{Cov(\hat{f}(x), f(x))}{\sigma_{\hat{f}(x)}\sigma_{f(x)}} \tag{6.3}$$

where $Cov(\hat{f}(x), f(x))$ is the covariance of the reference and reconstructed images, $\sigma_{\hat{f}(x)}$ is the standard deviation of the reference image, $\sigma_{f(x)}$ is the standard deviation of the reconstructed image. The value of CC is between -1 and 1 where 1 is the total positive linear correlation, 0 is no linear correlation and -1 is total negative linear correlation.

6.3.2 4D Extended Cardiac-Torso (XCAT) Phantom

In this work, datasets from the digital 4D Extended Cardiac-Torso (XCAT) Phantom [115] were used in the experiments. The thorax anatomy structure of the phantoms were selected to show the performance of the proposed algorithm to reconstruct lung images. Three datasets were used for the training and testing purposes. The details of experiments with each dataset are explained in the following sections.

Training dataset

The first dataset used to train the hyper-parameters for the AwPCSD reconstruction algorithm using the proposed computer-aided training approach is the XCAT phantom as shown in figure 6-2. The chosen voxel size is $128 \times 128 \times 128$.

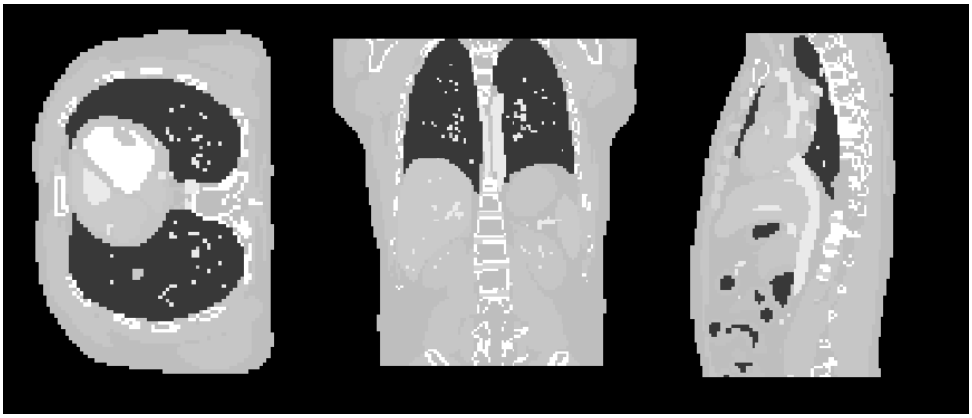


Figure 6-2: Cross-sectional slices of the XCAT phantom used for the training of hyper-parameters in transverse, coronal and sagittal planes.

Poisson and Gaussian noise [135], [84] was added to the input projection data of this data set to simulate a realistic noise. The case of default noise in the experiment was a combination of Poisson noise with maximum photon count of 60,000 and the Gaussian noise with mean and standard deviation of 0 and 0.5, respectively.

In this experiment, the following general parameters for the algorithm are defined as: maximum number of iteration = 50, maximum number of generation of ant colony = 10, number of ants in a colony = 50, evaporation rate = 1. The configurations for five hyper-parameters in this experiment are displayed in the table 6.1.

Table 6.1: Values of hyper-parameter configurations for this study

Hyper-parameters	Values
Data-inconsistency-tolerance parameter (ε)	0,50,70,100,200,500,2×10^3,1×10^4,1×10^5,5×10^5
TV sub-iteration number (ng)	2,4,6,8,10,12,14,16,18,20,22,24,26,28,30
Relaxation parameter (β)	1
Reduction factor of relaxation parameter (β_{red})	0.99
Scale factor for adaptive-weighted TV norm (δ)	0.0213

The values of three hyper-parameters are fixed, as explained in the method section earlier. These values remain constant for all hyper-parameters configurations. There are 10 values for ε and 15 values for ng. In total, there are 150 hyper-parameter configurations to be optimised by the proposed algorithm. Each ant then chooses the values of the remaining hyper-parameter, ε and ng, to search for the optimal hyper-parameter configuration for a given data.

The performance of the proposed algorithm was evaluated in a limited data scenario by using the data set with 50 projection views, equally sampled from a 360° with an increment of 7.2° between each angle.

6.3.3 Performance evaluation

To evaluate the performance, the reconstructed image obtained from the optimal hyper-parameters from the proposed method was compared with the results from other 3 cases. The first case is the reconstruction result from the CGLS algorithm. The second case is the AwPCSD reconstruction algorithm using the non-optimal set of hyper-parameters. The non-optimal set represents the way a user who is not familiar with the TV-based CT reconstruction would have chosen this set of hyper-parameters. The last case to compare is a cross-validation method, which is explained in this section.

The cross-validation method

The cross-validation method [5] [126] is a technique that is used to evaluate predictive models by splitting the original data into training and testing sets. Typically, the implementation of the

cross-validation method for the hyper-parameter calibration is prohibitively time consuming. The method is implemented for a comparison in this work to evaluate the performance of the proposed method in terms of the quality of reconstruction, as well as the computational time.

The implementation of cross-validation technique to select the hyper-parameter in this work is shown in figure 6-3 and explained as following. In one trial of the cross-validation technique, one projection is taken out from the available stack of projection data and is used as a testing data. The remaining projections are used as a training set. The reconstruction using the AwPCSD algorithm is implemented on the training set in parallel, using each one of the hyper-parameter configurations. The RMSE error is measured between the projected data from the reconstructed image of each hyper-parameter configuration and the testing data. Then, the method moves on to the next trial, where the next successive projection data is used as a testing data instead of the current one. The same process is repeated again until the last available data is used as a testing data, i.e. the total number of trial equals to the number of projection data. The average error of each hyper-parameter configuration is computed from the errors collected from all trials. The best parameter configuration is the one with the lowest average error and the worst parameter configuration is the one with the highest average error.

As the results of the cross-validation method is evaluated on a testing data, which is not included in the data used to produce the results, the cross-validation method is thus a good benchmark to compare with the proposed algorithm.

The proposed algorithm, as well as the other 3 methods, were implemented with the simulated XCAT phantom projection data, following the experimental setting as explained earlier. Figure 6-4 presents the scores computed as correlation coefficients of the reconstructed images using all the hyper-parameter configurations. The plot in figure 6-4 are scores of the hyper-parameter configurations as chosen by all members of ants in the last iteration. These scores are accumulated over the entire implementation of the proposed ACO hyper-parameter selection algorithm. The one with the highest score is chosen as an optimal set of hyper-parameters. The optimal set of hyper-parameters found by the proposed algorithm and the cross-validation method, as well as the non-optimal setting are displayed in the table 6.2. Note that for the CGLS algorithm, no hyper-parameter is required. The maximum number of iteration for 3 cases is specified at 50. However, the CGLS algorithm converges and stops at the iteration number 15 and the AwPCSD algorithm with the non-optimal setting converges and stops at the iteration number 5.

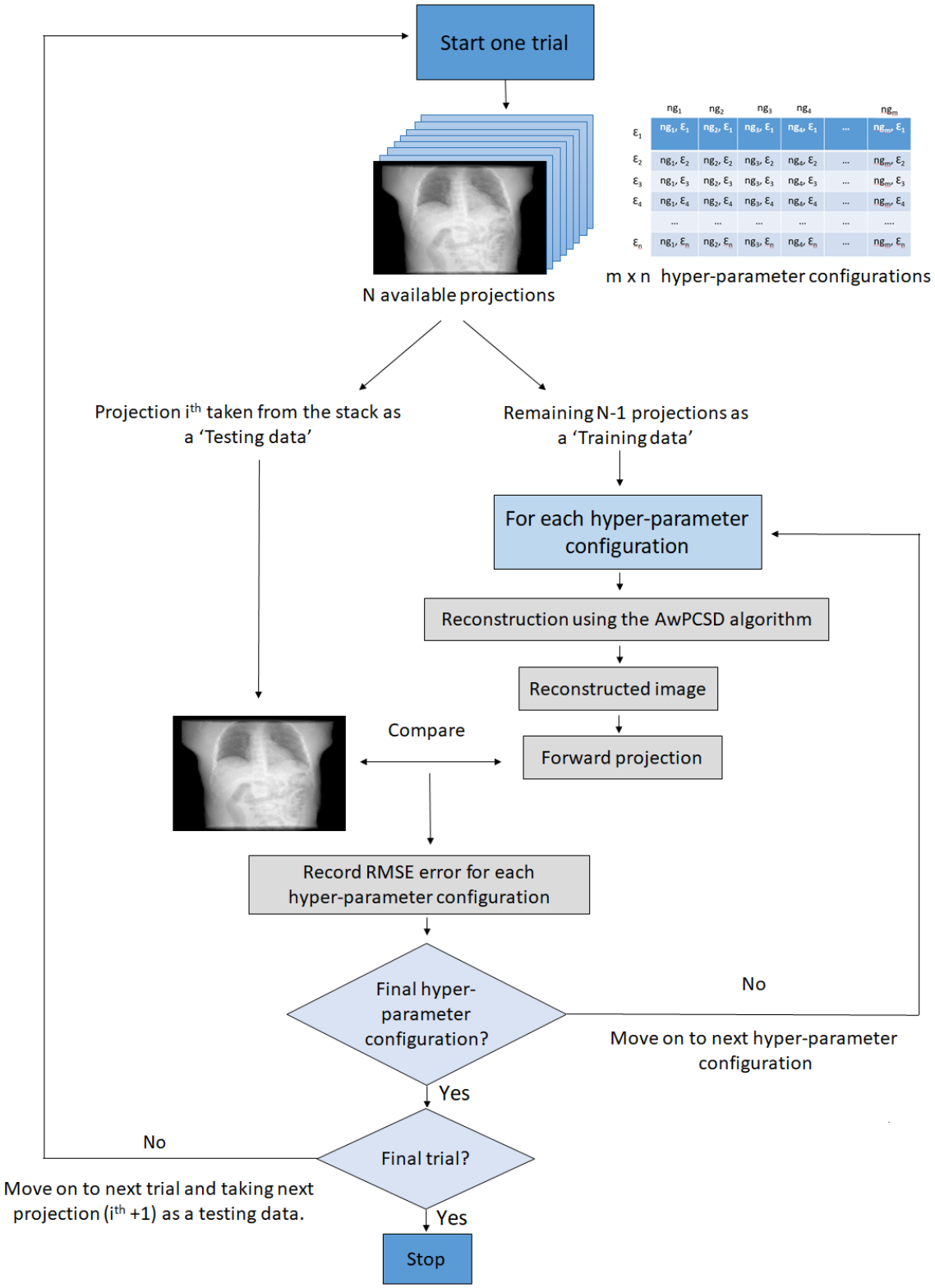


Figure 6-3: The diagram of cross-validation algorithm

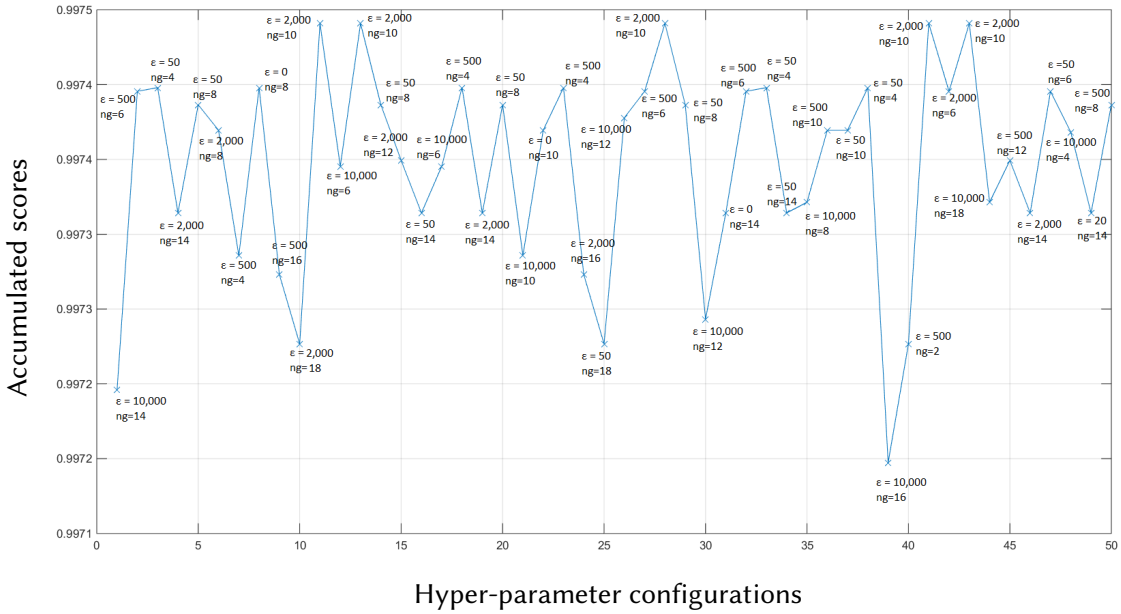


Figure 6-4: The plot of accumulated scores computed as correlation coefficients at the end of the implementation of the proposed ACO hyper-parameter selection algorithm. The values of each hyper-parameter configuration are annotated for each score in the figure.

Table 6.2: Different sets of hyper-parameters used to compare the performance of the proposed algorithm

Hyper-parameter selection methods	ε	ng	β	β_{red}	δ
Proposed algorithm	2,000	10	1	0.99	0.0213
Non-optimal setting	700	100	1	0.99	0.0213
Cross-validation	0	8	1	0.99	0.0213

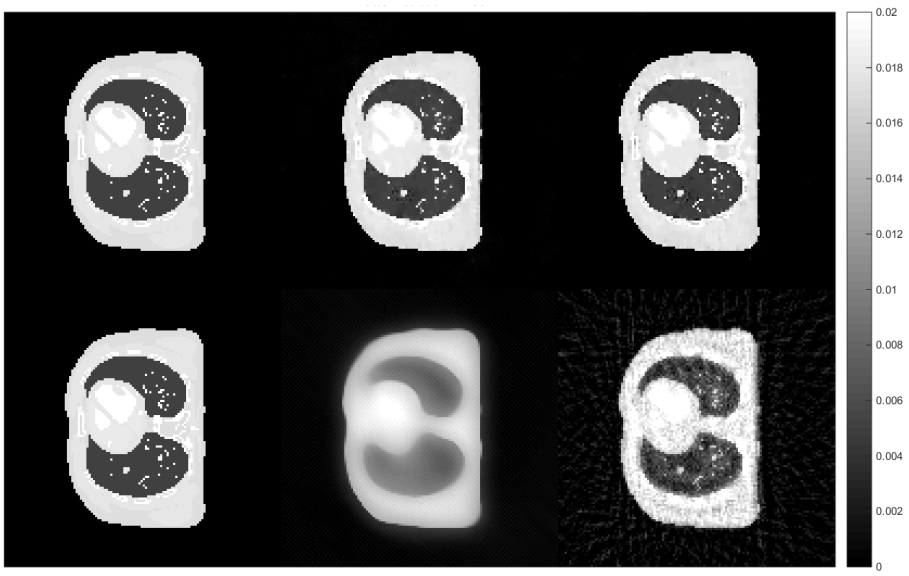
It is worth mentioning here that the $\varepsilon = 0$ for the case of cross-validation does not mean that the algorithm was able to achieve no error between the predicted and observed projection data. In this case, what happened was that the algorithm aimed to reach the zero error point, when the hyper-parameter configuration under study contains $\varepsilon = 0$, but was forced to stop due to the maximum number of iteration was reached first. Thus, the hyper-parameter configurations which contain $\varepsilon = 0$ was evaluated based on this attempt, in combination with the performance as affected by the other hyper-parameters.

The testing computer used for the experiment is Intel Core i7-4930K CPU @3.40GHz with 32 GB RAM. The single GPU in use is NVDIA GeForce GT 610. Comparing the computational time for all 4 cases, the CGLS algorithm is the one with the shortest time as it only takes a couple of seconds to finish its 15 iterations. For the AwPCSD algorithm with the non-optimal setting of hyper-parameters, it takes approximately 2 minutes for its implementation. The hyper-

parameter learning process using the proposed algorithm takes approximately 1.45 hours to finish. The cross-validation algorithm is the one with the longest computational time. It takes approximately 47.15 hours to finish the entire process.

Cross-sectional slices of the reconstruction results from all cases are shown in figure 6-5, in comparison with an exact phantom image.

(a) Exact image (b) Proposed algorithm (c) Cross-validation



(a) Exact image (d) Non-optimal setting (e) CGLS

Figure 6-5: Cross-sectional slices of reconstructed images from 50 projection views obtained from 4 cases (a) Exact image (b) the proposed algorithm (c) Cross-validation (d) Non-optimal setting (e) CGLS. The display window is [0-0.02].

From a visual inspection of figure 6-5, the reconstructed image from the proposed algorithm and the cross-validation method are rather similar to each other. Not much outstanding difference can be observed in the images from both cases. However, they are very similar the the exact phantom image, as they contain sharper edges compared to the blurry result from the non-optimal setting. Even though the CGLS algorithm is able to recover small features as well as edges, the image is relatively noisy compared to the result from the proposed algorithm and the cross-validation method.

To make a comparison clearer, we analysed one-dimensional profile plots of all the results along an arbitrary row of a cross-sectional slice as shown in figure 6-6. The 1D plot of all the results is shown in figure 6-7, in comparison with the reference exact phantom.

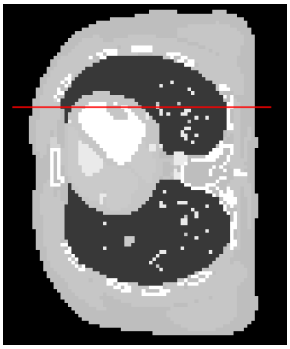


Figure 6-6: The image profiles of all the results along the horizontal line are plotted.The display window is [0-0.02].

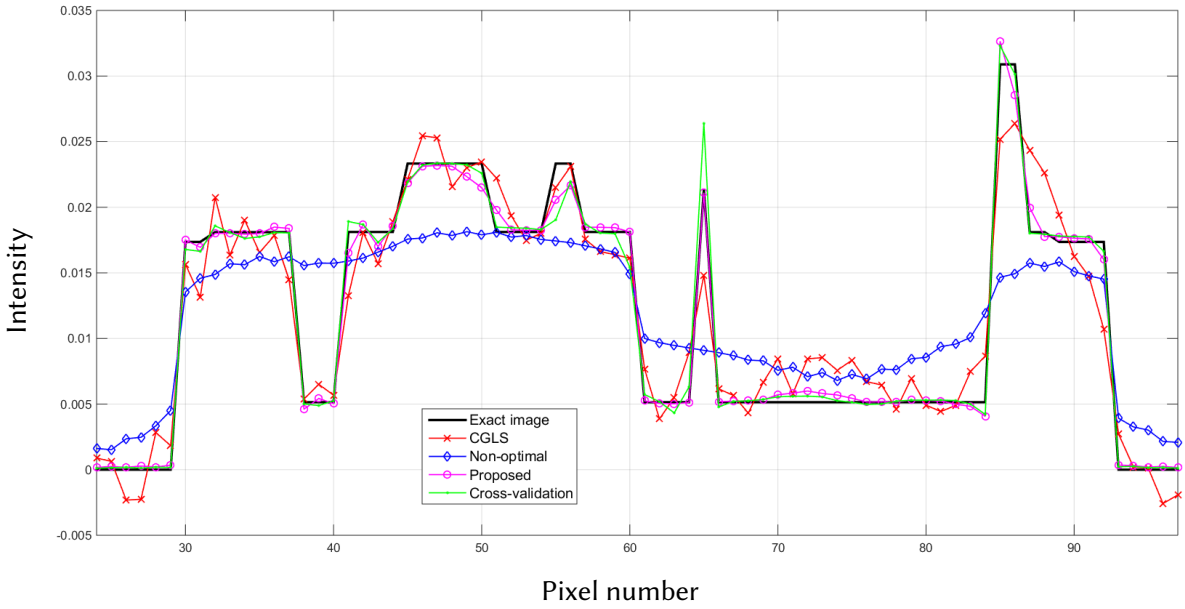


Figure 6-7: One-dimensional profiles plot of all the results from 4 cases, in comparison with the exact image.

According to the plots in figure 6-7, the one-dimensional profiles from the proposed ACO algorithm and the cross-validation method are the most aligned with the profile from the exact image. This is particularly true around the edges area, where the abrupt changes occur. In some areas, the profile from the proposed algorithm is more aligned to the exact image than that of the cross-validation method. But there are also some other areas that the cross-validation method is better aligned. Hence, the results from these two algorithms do not have any outstanding difference still. In line with the visual inspection of the figure 6-5, the profile from the AwPCSD algorithm using the set of non-optimal hyper-parameter setting shows that the algorithm failed to recover the edges information of the image. This can be seen between pixel number 30 to 60, where the profile plot is rather flat. The result from the CGLS is middle ground between the first 2 cases. Although the CGLS algorithm is able to recover most of the image features, the result is much noisier compared to the proposed ACO algorithm.

The result from the AwPCSD algorithm with the non-optimal setting can be improved further by re-selecting the values of hyper-parameters and implement the algorithm again. It is a time-consuming and tedious process and there is no way to guarantee that the chosen hyper-parameters will be the optimal ones. This highlights the significance and advantage of having the proposed computer-aided hyper-parameter selection algorithm, which helps to save time and resource of the user to find an optimal set of hyper-parameters.

Application to other cases

To evaluate the robustness of the set of hyper-parameters obtained from the proposed algorithm in different scenarios, the optimal set of hyper-parameters obtained from the proposed algorithm was used to reconstruct images from projection data with 3 different levels of noise. The projection data simulated using the digital XCAT phantom is added the Poisson and Gaussian noise, as explained in the following:

- **Noise 1 case**: Poisson noise = 30,000 maximum photon count and Gaussian noise with mean = 0, standard deviation of 1
- **Noise 2 case**: Poisson noise = 20,000 maximum photon count and Gaussian noise with mean = 0, standard deviation of 3
- **Noise 3 case**: Poisson noise = 10,000 maximum photon count and Gaussian noise with mean = 0, standard deviation of 5

The AwPCSD algorithm is used to reconstruct the images by taking each of the noise level case as an input. Cross-sectional slices of the reconstructed results in all cases are shown in figure 6-8.

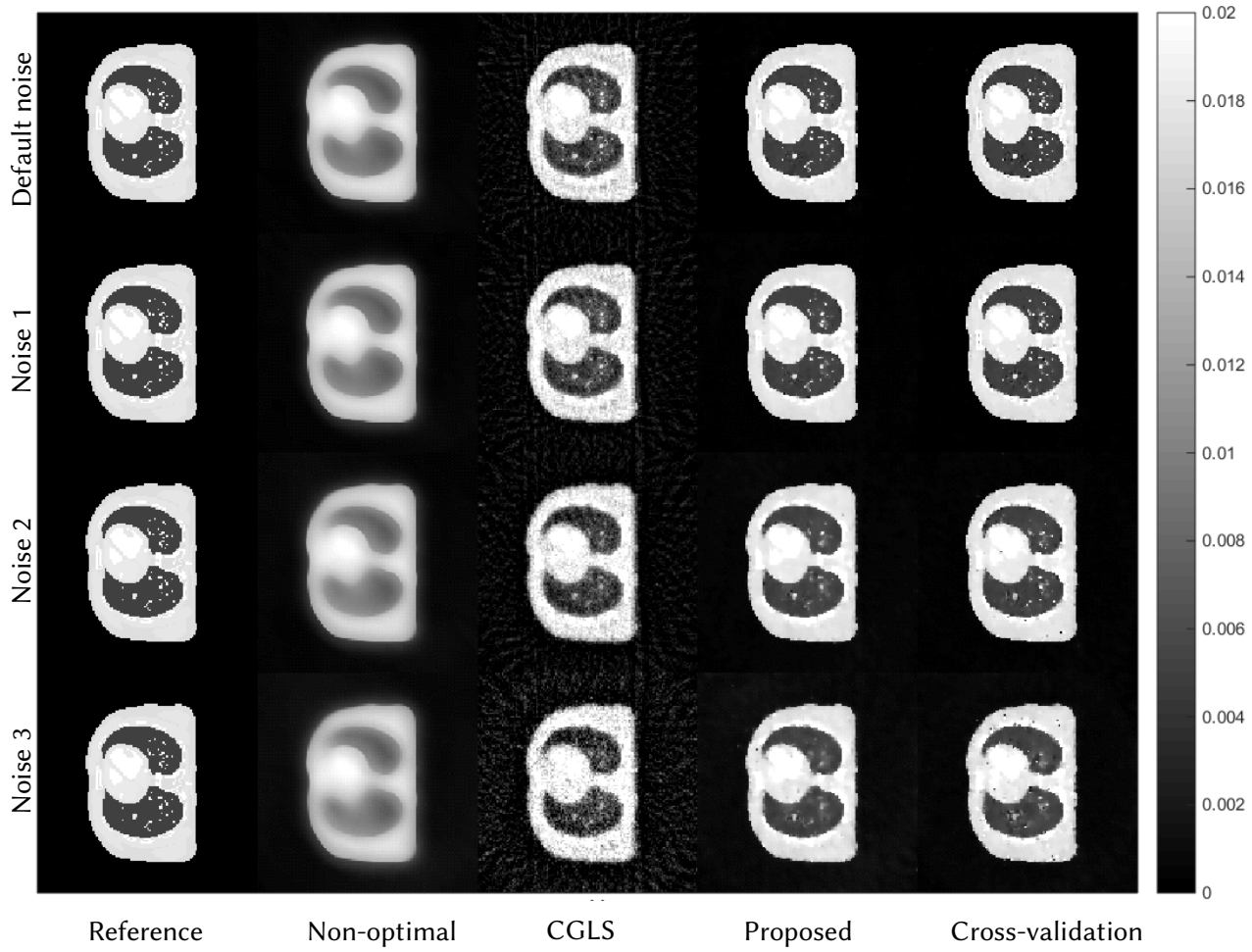


Figure 6-8: Reconstruction results from the projection data with different levels of noise. The display window is [0-0.02].

In order to quantify the difference between the results, the Universal Quality Index (UQI) [134], [144] values as well as the relative 2-norm errors in all different noise level cases, as computed using the below equations, are shown in table 6.3.

$$UQI = \frac{2cov(\mu,\mu_{true})}{\sigma^2+\sigma^2_{true}}\frac{2\bar{\mu}\bar{\mu}_{true}}{\bar{\mu}^2+\bar{\mu}^2_{true}} \tag{6.4}$$

$$\bar{\mu} = \frac{1}{Q}\sum_{m=1}^{Q}\mu_m, \quad \sigma^2 = \frac{1}{Q-1}\sum_{m=1}^{Q}(\mu_m-\bar{\mu})^2,$$

$$\bar{\mu}^{true} = \frac{1}{Q}\sum_{m=1}^{Q}\mu_m^{true}, \sigma^2_{true} = \frac{1}{Q-1}\sum_{m=1}^{Q}(\mu_m^{true}-\bar{\mu}_{true})^2,$$

$$cov(\mu,\mu^{true}) = \frac{1}{Q-1}\sum_{m=1}^{Q}(\mu_m-\bar{\mu})(\mu_m^{true}-\bar{\mu}_{true})$$

where μ and μ_{true} denote voxel intensity of reconstructed image and reference image, re-

spectively, m is the voxel index, Q is number of voxels, $cov(\cdot,\cdot)$, σ and $\bar{\mu}$ are covariance, variance and mean of intensities, respectively. The UQI is an image quality metric which is used to evaluate the degree of similarity between the reconstructed and reference images. The value ranges from zero to one, where the value closer to one suggests better similarity to the reference image.

$$relative\ \ error\ \ (e) = \frac{||image - image_{ref}||_2}{||image_{ref}||_2} \tag{6.5}$$

Table 6.3: Relative errors and UQI of image reconstruction results with XCAT Thorax phantom (Boldface numbers indicate the best result in each case)

Case	e_{cross}	$e_{proposed}$	$e_{non-optimal}$	e_{CGLS}	UQI_{cross}	$UQI_{proposed}$	$UQI_{non-optimal}$	UQI_{CGLS}
Default	**5.25%**	6.70%	29.13%	19.26%	**0.9970**	0.9959	0.9322	0.9749
Noise 1	**6.95%**	8.97%	29.62%	20.64%	**0.9959**	0.9938	0.9293	0.9711
Noise 2	**11.23%**	13.32%	31.42%	23.22%	**0.9905**	0.9857	0.9182	0.9631
Noise 3	18.37	**18.01%**	32.86%	31.83%	**0.9737**	0.9717	0.9071	0.9330

According to figure 6-8 and table 6.3, the results from the AwPCSD with non-optimal setting are clearly the worst among the others. This is because the cross-sectional slices are rather blurry and the relative errors are the biggest with the lowest UQI values in all cases. The CGLS results are better than those of the non-optimal setting but still corrupted by noise, also worsen with an increase of noise levels. Quantitatively, the relative errors and the UQI values show that the cross-validation method is slightly better than the proposed algorithm in all cases apart from relative error in the noise 3 case. Although the results from the proposed algorithm is slightly inferior than the cross-validation method, the fact that the cross-validation method is much more computationally expensive in term of the running time makes the method impractical in real use. The cross-validation method is implemented here for a comparison purpose. The point we are making in this experiment is to prove that the proposed algorithm is able to achieve almost as good quality of the result as that of the cross-validation method, but in a much more feasible time frame.

The experiment in this section proves that the same set of hyper-parameters from the proposed method is robust against an increase of noise in the projection. The reconstructed images from the proposed algorithm still able to maintain a superior quality over almost all other methods in all the noise cases.

Further analysis with multiple arrangements of angles used to acquire the projection data was tested to ensure that the results are not unique to only one specific angle arrangement. The projection data used in the training of hyper-parameters contains 50 projection views, equally sampled from a 360° angle with an increment of 7.2° between each angle. This hyper-parameter setting was tested with four different angle arrangements with fixed number of projections. The

first two angle arrangements are the projection data collected over 360° with 7° and 5.9° increments, respectively. The other two arrangements are collected over 180°, with 3.6° and 3.52° increments. Cross-sectional slices of the reconstructed images with different angle arrangements using the same set of hyper-parameters from the training stage of the proposed algorithm is shown in figure 6-9.

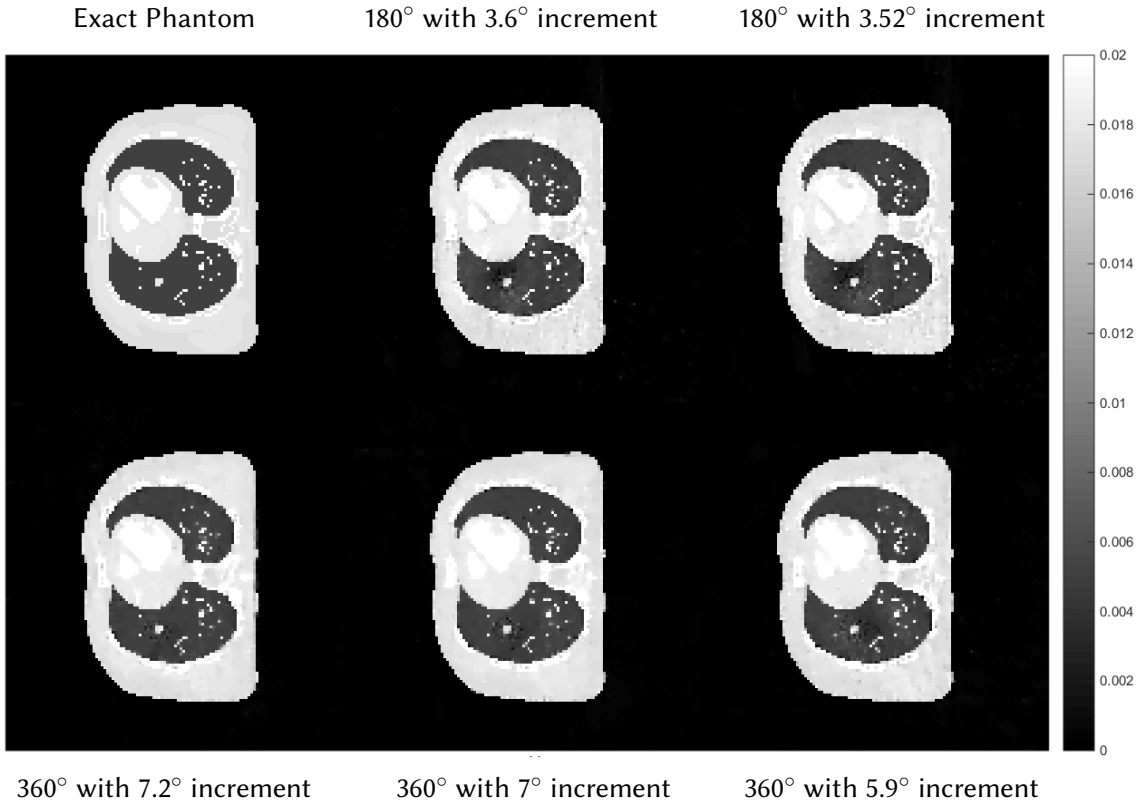


Figure 6-9: Cross-sectional slices of the reconstructed images with different angle arrangements. The display window is [0-0.02].

The difference images between the cross-sectional slices in figure 6-9 and the exact phantom are displayed in figure 6-10 to better observe the differences in each case.

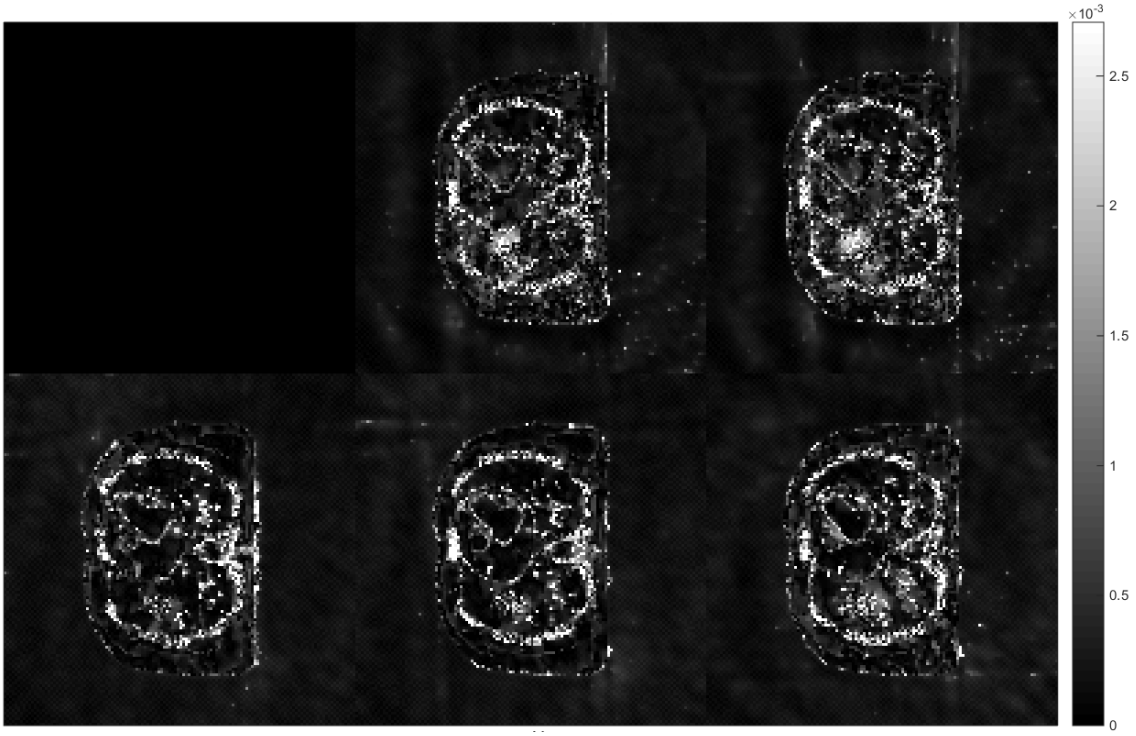


Figure 6-10: The difference images in each angle arrangement between the exact XCAT phantom and the reconstructed images using the set of hyper-parameters from the proposed algorithm. The display window is [0-0.02].

The results in figure 6-9 and the difference images in 6-10 show that the hyper-parameter setting obtained from the proposed algorithm is able to reconstruct the same quality of image even when the angle arrangement of the projection data was changed.

Testing datasets

In the second part of experiments, the robustness of the proposed algorithm was evaluated by applying the trained hyper-parameter setting to the reconstruction of different samples. For this purpose, two different XCAT phantom datasets were simulated with different parametrisation from the training dataset.

The first dataset is a male phantom with the chosen voxel size of $128 \times 128 \times 70$. The second one is a female phantom with some modifications of general parameters used to generate the phantom. The detail of the different modifications between these two phantoms is shown in the table 6.4. Cross-sectional slices of the two phantoms are shown in figure 6-11.

Table 6.4: The parametrisation details of the two phantoms

Parametrisation details	Male	Female
motion option	beating heart only	respiratory only
length of beating heart cycle	1 sec	5 secs
starting phase of the heart	0.0	0.4
wall thickness for the left ventricle(LV)	non-uniform	uniform
LV end-systolic volume	0.0	0.5
start phase of the respiratory	0.0	0.4
anteroposterior diameter of the ribcage, body and lungs	0.5	1.2
heart's lateral motion during breathing	0.0	0.5
heart's up/down motion during breathing	2.0	3.0
breast type	prone	supine
factor to compress breast	half compression	no compression
thickness of sternum	0.4	0.6
thickness of scapula	0.35	0.55
thickness of ribs	0.3	0.5
thickness of backbone	0.4	0.6

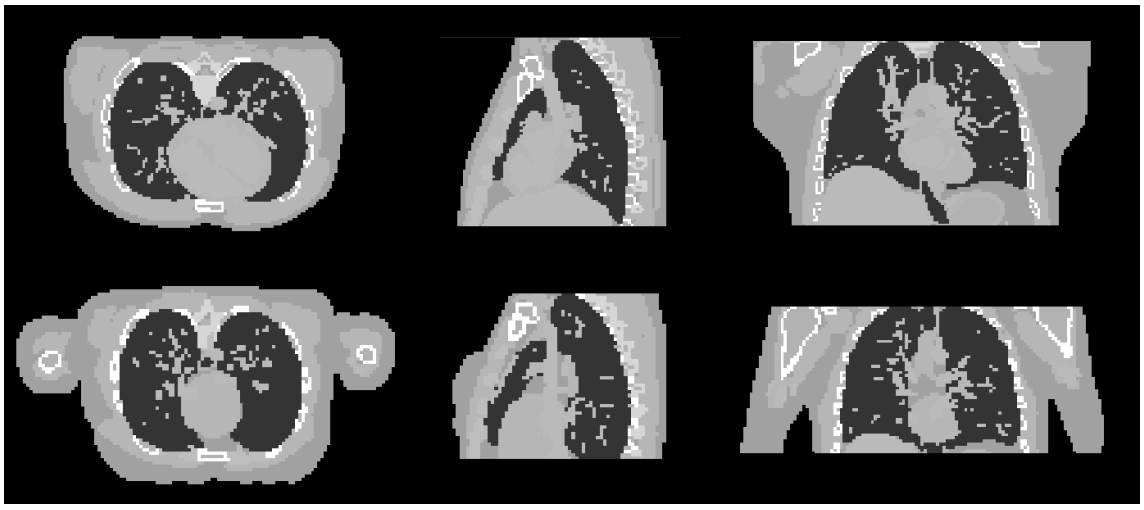


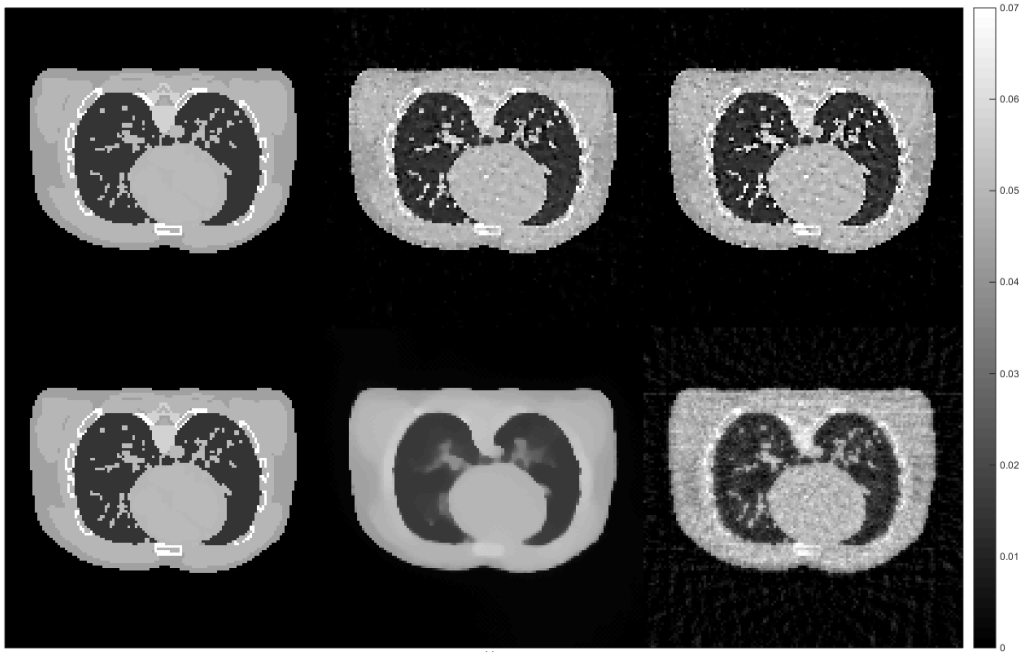
Figure 6-11: Cross sectional slices of the male (top row) and female (bottom row) phantoms in the three axes.

In this experiment, the experimental setting used in the previous experiment with the first set of XCAT phantom remains the same, i.e. the default case is 50 projection views, equally sampled from a 360° with Poisson noise of 60,000 maximum photon count and Gaussian noise with mean and standard deviation of 0 and 0.5, respectively.

In the same way, the same sets of hyper-parameters used in the previous experiment as

displayed in the table 6.2 were used to reconstruct images of the testing datasets, as well as the CGLS algorithm. The results of the male phantom are shown and discussed first. The cross-sectional images are shown in the figures 6-12.

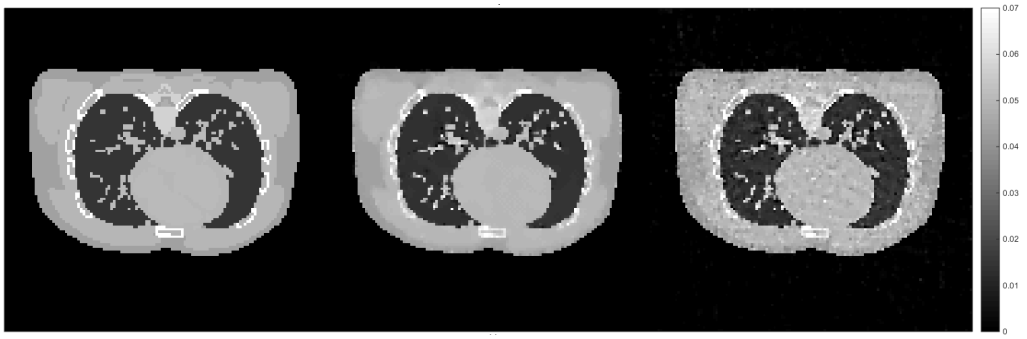
(a) Exact image (b) Proposed algorithm (c) Cross-validation



(a) Exact image (d) Non-optimal setting (e) CGLS

Figure 6-12: Cross-sectional slices of the reconstructed images from the male phantom using the same sets of hyper-parameter settings and the CGLS algorithm as applied to the training dataset previously. The display window is [0-0.07].

According to figure 6-12, the reconstruction using the set of hyper-parameters obtained from the training dataset still able to produce good quality of image, comparing to other methods. We then implement the proposed method directly to the projection simulated from the male phantom to further analyse the difference between these 2 cases. The cross-sectional slices are shown in figure 6-13. The relative errors and the UQI, as well as the sets of hyper-parameters used in each method are presented in table 6.5.



(a) (b) (c)

Figure 6-13: Cross-sectional slices of (a) the exact image ,(b) the reconstruction from the proposed method directly implemented on the male phantom data, (c) the reconstruction using the set of hyper-parameters from the training dataset. The display window is [0-0.07].

Table 6.5: Relative errors and UQI of image reconstruction results from the male phantom using each set of hyper-parameter and the CGLS algorithm. (Boldface numbers indicate the best result)

Sets of hyper-parameter/method	**Relative error (%)**	**UQI**	ε	ng
Proposed method from the training dataset	8.2668%	0.9946	2,000	10
Proposed method with the male phantom	**5.3356%**	**0.9980**	70	22
Cross-validation from the training dataset	10.6377%	0.9909	0	8
Non-optimal setting	12.7896%	0.9863	700	100
CGLS	15.9408%	0.9798	N/A	N/A

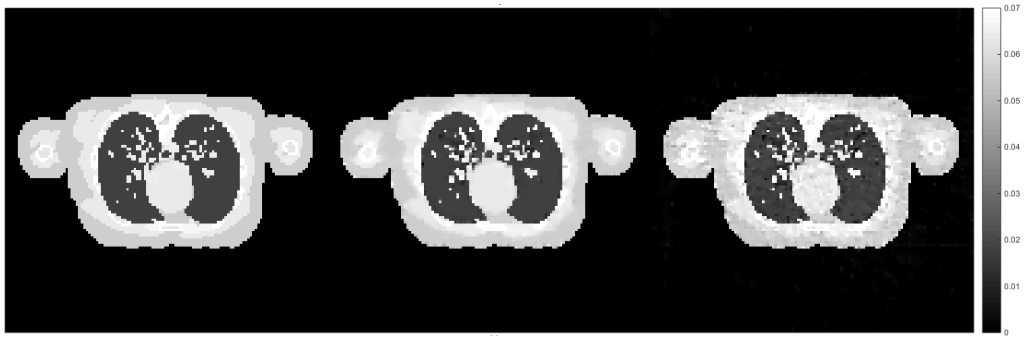
According to figure 6-13 and table 6.5, it can be concluded that the set of hyper-parameters obtained from the proposed algorithm with the training set can be applied to different imaging sample and produce the result which is superior to other methods. However, the set of hyper-parameters obtained from directly implementing the proposed algorithm with the male phantom projection shows even better result. This experiment proves that the selection of hyper-parameter for the TV regularisation algorithm, specifically for the AwPCSD algorithm, is data-specific. The optimal set of hyper-parameters from one training dataset can still be applied to different image sampling with similar context. However, the optimal result might not yet be achieved. It is significant to fine-tune the hyper-parameters, in order to obtain the optimal result for a given data. This is the advantage that the proposed algorithm offers, to avoid the tedious process of manual hyper-parameter tuning.

The same pattern of experiment is performed on the female phantom and the results, as presented in the figures 6-14,6-15 and table 6.6, confirm the conclusion stated above.

(a) Exact image (b) Proposed algorithm (c) Cross-validation

(a) Exact image (d) Non-optimal setting (e) CGLS

Figure 6-14: Cross-sectional slices of the reconstructed images from the female phantom using different method and hyper-parameter settings. The display window is [0-0.07].



(a) (b) (c)

Figure 6-15: Cross-sectional slices of (a) the exact image, (b) the reconstruction from the proposed method directly implemented on the female phantom data, (c) the reconstruction using the set of hyper-parameters from the training dataset. The display window is [0-0.07].

Table 6.6: Relative errors and UQI of image reconstruction results from the female phantom using each set of hyper-parameter and the CGLS algorithm. (Boldface numbers indicate the best result)

Sets of hyper-parameter/method	Relative error (%)	UQI	ε	ng
Proposed method from the training dataset	7.5255%	0.9963	2,000	10
Proposed method with the female phantom	**5.5501%**	**0.9982**	70	18
Cross-validation from the training dataset	8.8493%	0.9950	0	8
Non-optimal setting	16.7038%	0.9800	700	100
CGLS	19.3525%	0.9752	N/A	N/A

6.3.4 RANDO anthropomorphic head phantom

In this section, the set of hyper-parameters obtained from the proposed algorithm with the training dataset is tested on the real data, the RANDO anthropomorphic head phantom. The projection data used for the experiment in this section is the measured projection data from the RANDO head phantom [1], which were provided by North Western Medical Physics at The Christie Hospital in Manchester. The dataset contains 360 X-ray projection images, approximately evenly spaced over a range of -100° to +100° . The projection images were acquired at 100kV, 10mA and 10ms per projection, with total imaging dose of approximately 1.5mGy. The size of each projection image is 512×512 pixels of dimension $1 \times 1mm$, which is a down-sampled size from the system maximum resolution of 1024×1024. An example of one measured projection image is shown in figure 6-16.

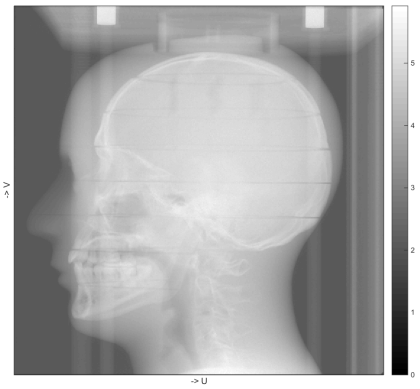


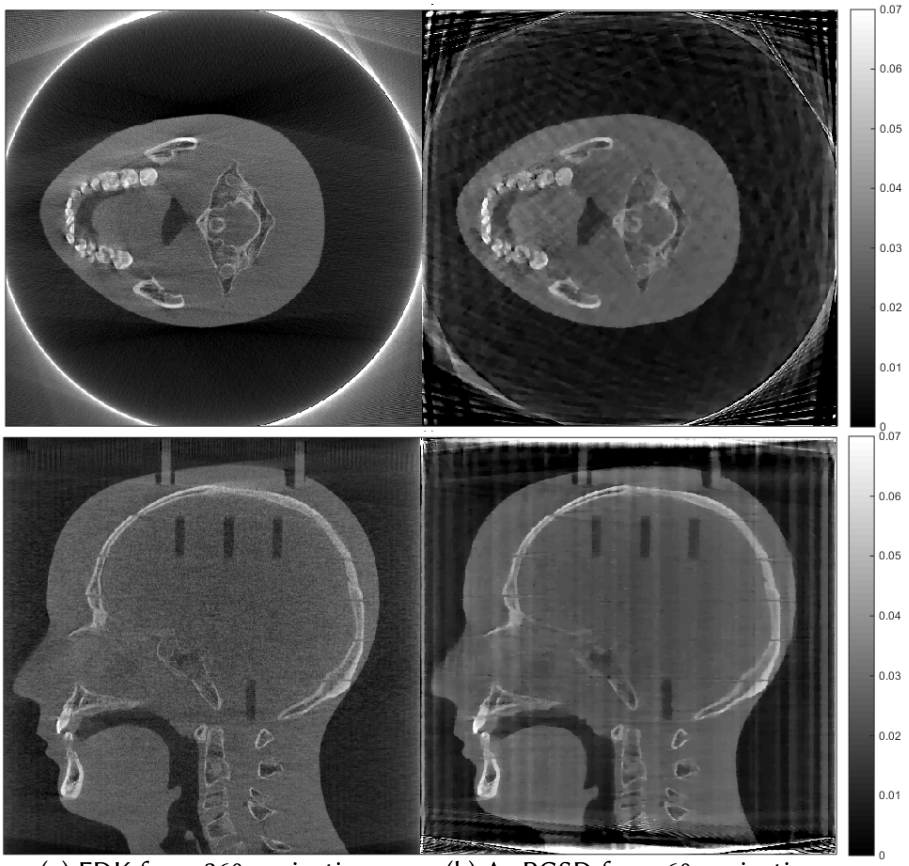
Figure 6-16: One measured projection image from the RANDO head phantom

The AwPCSD reconstruction algorithm is applied to the projection of Rando head phantom using the trained hyper-parameters from the first XCAT thorax. The purpose of the experiment is to observe if the trained hyper-parameters from one training set can be applied to different

[1]The Phantom Laboratory, Salem, NY, USA.

imaging sample in a different context, i.e. from the thorax phantom to the head phantom.

The number of projection for the experiment is sub-sampled from 360 to 60, equally sampled over 360°. Since the Rando head projection is the real dataset, there is no exact phantom image to compare the result with. The reference image used for a comparison is then obtained using the FDK algorithm with full projection data (360°). The computational time taken for 50 iterations of the AwPCSD algorithm for 60 projection images using the same testing computer is ≈ 1 hour 4 minutes. The cross-sectional slices of the result is shown in figure 6-17, in comparison with the FDK result.



(a) FDK from 360 projections (b) AwPCSD from 60 projections

Figure 6-17: Cross-sectional slices of the reconstructed images from the RANDO head phantom dataset using (a) the FDK algorithm with 360 projection images, (b) the AwPCSD algorithm using the hyper-parameter obtained from the proposed algorithm with the training dataset.

It can be seen from the figure 6-17 that the AwPCSD algorithm using the hyper-parameters from the training set still able to produce an image with reasonable quality. However, the quality of the reconstructed image is much inferior to the result from the FDK algorithm with full set of projection. The potential reasons could be because the number of projections for the AwPCSD algorithm is much lower and the hyper-parameters are not yet specifically calibrated for the Rando head phantom. The result could be further improved by implementing the proposed algorithm on the Rando head projection data to select an optimal set of hyper-parameters.

Testing the RANDO head phantom data with the ACO hyper-parameter selection algorithm

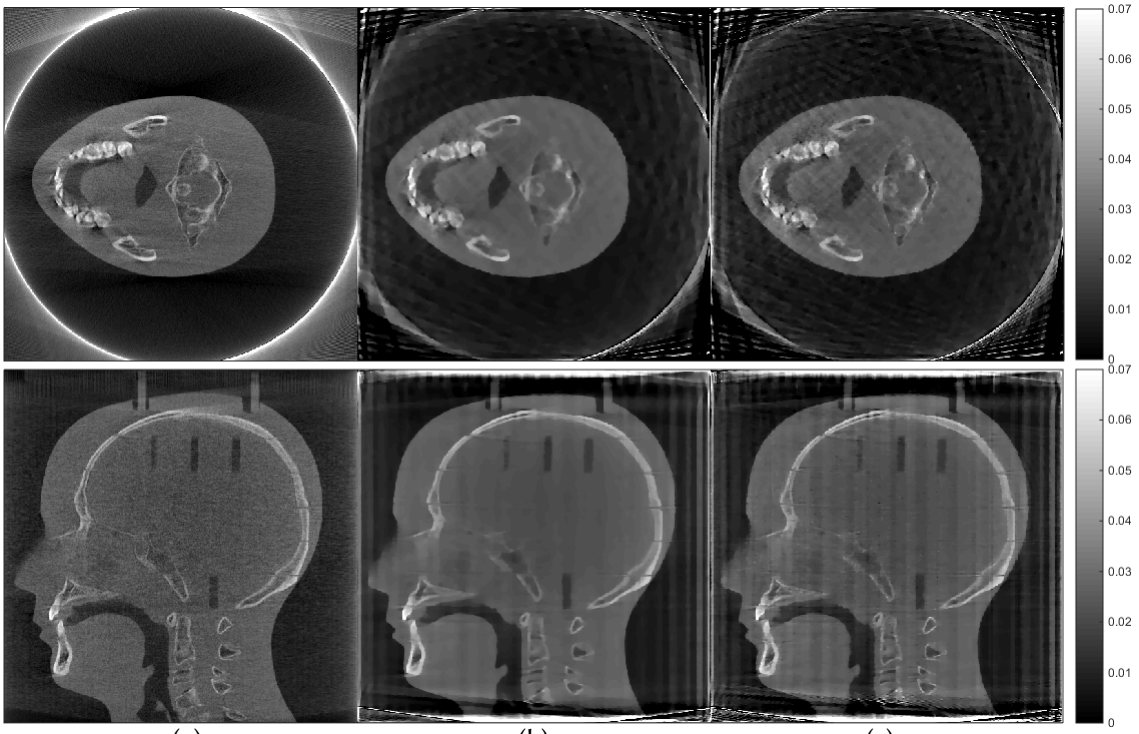
In this section, the ACO hyper-parameter selection is implemented directly on the projection data from the RANDO head phantom. This is to evaluate the performance of the ACO algorithm with the real projection data and to compare the result with the reconstruction from the trained hyper-parameters from the previous data set. The number of projection used is 60 projection views. The reference image for the algorithm is the result from the FDK algorithm with full projection data (360°). The number of image voxel size is $512 \times 512 \times 512$. Due to the massive size of the dataset, the computational time taken to finish the entire process of the ACO algorithm using the same setting as that of the XCAT phantom is $\approx$ 17 days. With the same initial hyper-parameter values, table 6.7 summarises the best hyper-parameter configurations, as well as the UQI and relative errors from different experiments. The reconstructed images from the two cases, as well as the reference FDK image, are shown in figure 6-18.

Table 6.7: Best hyper-parameter configurations as found by the ACO hyper-parameter selection algorithm from different experiments

Experiments	TV sub-iteration number (ng)	Data-inconsistency-tolerance parameter (ε)	Relative error (%)	UQI
The ACO method with the XCAT phantom	10	2,000	77.8337	0.3408
The ACO method with the RANDO phantom	30	0	**74.9196**	**0.3601**

According to the results of this experiment with the RANDO head data, the following points can be concluded. Firstly, the results prove again the point we have made earlier that the reconstruction using the set of hyper-parameters obtained from directly implementing the ACO method are better than reconstruction using the trained hyper-parameters from the other data set. Although the difference between two cases are relatively small in this experiment, it is still noticeable. However, the computational time taken to implement the algorithm to the real data with massive geometry is extremely long, for the current version of the algorithm. Therefore, it is not always practical to apply the algorithm directly to the big size data. The experiment in this section gives promising results, however, that if we were to use the non-optimal set of hyper-parameters for the reconstruction, the fundamental structure of the algorithm is able to achieve an image with a reasonable quality. The trained hyper-parameters from the RANDO head phantom data are also promising to be able to use with other data set with the same context, i.e. the brain projection data. This will save time for the training of the other brain projection data, which could possibly take very long computational time as well.

There are several alternatives that could be implemented to reduce a computational time of



(a) (b) (c)

Figure 6-18: Cross-sectional slices of the reconstructed images from the RANDO head phantom dataset using (a) the FDK algorithm with 360 projection images, (b) the AwPCSD algorithm using the trained hyper-parameters from the RANDO head phantom data, (c) the AwPCSD algorithm using the trained hyper-parameters from the XCAT phantom data

the ACO hyper-parameter selection algorithm. First option, the algorithm can be modified to use with the multi-GPU, which should drastically accelerate the computational time. Secondly, the voxel size of the reconstructed image can be specified a smaller value, for which the quality of the image might be compromised.

6.4 Summary

In this chapter, the computer-aided hyper-parameters selection algorithm for limited data CT reconstruction using the TV regularisation algorithm is proposed. In the proposed algorithm, the AwPCSD algorithm is used as a reconstruction algorithm. The ACO approach is employed to select the optimal set of hyper-parameter for the reconstruction with the AwPCSD algorithm, which is crucial for the reconstruction result.

Initially, the ranges of hyper-parameter values are specified. The proposed algorithm searches through all possible configurations via a colony of ants and evaluate each configuration based on the score obtained from the comparison between the reconstructed image and the reference image. The pheromones are left for all configurations according to the scores, to attract ants in the next generation. At the end of the implementation, the set of hyper-parameters with the highest score is considered as the optimal set for a given projection data. The implementation of the proposed algorithm is fully aided by computer, without the need of human intervention between the process.

The experimental results showed that the results from the proposed algorithm are superior to the results from CGLS algortihm and the AwPCSD algorithm using the non-optimal hyper-parameter setting. Although the results of the proposed algorithm are slightly inferior to those of the cross-validation method as measured by the quantitative metrics, the computational time of the proposed algorithm is much shorter, approximately more than 10 times quicker than the cross-validation method. Furthermore, the optimal set of hyper-parameters from the training data is robust against an increase of noise in the projection data. The reconstructed images from the proposed algorithm still able to maintain a superior quality over almost all the methods in all the noise cases.

In addition, the optimal set of hyper-parameters from one training dataset can still be applicable to different imaging samples with similar context. Depending on the requirement of users in term of imaging quality, the better result can be achieved by directly apply the proposed algorithm to the data. This is the advantage that the proposed algorithm offers, to avoid the tedious process of manual hyper-parameter tuning.

Chapter 7

Efficient hyper-parameter selection in Total Variation based CT reconstruction using Freund and Shapire Hedge algorithm

7.1 Introduction

From the results of the hyper-parameter selection algorithm using the ACO method presented in the previous chapter, there are two main points that could be improved for a better hyper-parameter selection method. The first point is that the results from the ACO algorithm are slightly inferior to those of the cross-validation method, in terms of the quantitative measures using the relative error and the UQI.

The second point is the requirement of the reference image for a comparison during the implementation. The reference image can either be an exact phantom image or a result from the FDK algorithm with full set of projection data. This would be problematic in case of the limited angular real projection data, where there is no exact phantom image available and only projection in certain angular range are available. The FDK result from an incomplete set of projection is not a good quality image. Hence, it is not practical to implement the hyper-parameter selection algorithm using ACO method in this scenario.

In this chapter, we take the motivation to improve the previously presented algorithm by considering the hyper-parameter selection problem from a new perspective, which combines the Hedge method of Freund and Shapire [46], [6] with the AwPCSD reconstruction algorithm.

The algorithm presented in this chapter combines the hedge algorithm of Freund and Shapire with sequential reconstruction using the AwPCSD algorithm based on successive incorporation of new projections. A probability distribution on the set of potential hyper-parameters is updated with every new projection. The hyper-parameter with the highest probability is selected for the reconstruction. The main result of this method is that choosing the value of hyper-

parameters using the probability mass after a certain number of steps provides a prediction error which is almost as accurate as the prediction error incurred by the best predictor.

The hyper-parameter selection algorithm using Freund and Shapire's hedge algorithm does not require a reference image because the comparison during the implementation of the algorithm is performed in the projection domain. Therefore, the algorithm is practical for real projection data with limited angles.

7.1.1 The Hedge algorithm

Firstly, the concept of Hedge algorithm by Freund and Schapire [46] is explained in this section. It is the online prediction model to solve dynamic allocation problems such as gambling, multiple-outcome prediction, repeated games and prediction of points in $\mathbb{R}^n$. With the word 'online', it means that a prediction model learns as it goes through experience when more aspects of the problem are observed. In each step, the prediction of the next observation is made and used to compute a loss that can be incorporated into the update rule of Freund and Shapire by using an exponential factor [6].

In the context of the hyper-parameter selection, the problem can be considered as the experts problem, which is the most well known problem in prediction theory [57]. An overview of the experts problem is given here in the horse-racing scenario, for instance. There are n experts who can give their advice about which horse is going to win in each round. The decision maker does not know which expert is the most knowledgeable about the horse-racing and needs to choose among all the possible n given experts. In the first round, the decision maker apportions his money equally among all the experts. In each round, the loss of each expert is computed and used to update the weight matrix for each expert using the multiplicative rule. The weight matrix reflects the performance of each expert. More money is allocated to the experts with higher weights for the next round of horse-racing. The scenario is repeated iteratively. The goal of the decision maker is to apportion his money in each race in a way that his total winnings will be close to what he would have won had he bet his money with the best expert in hindsight.

Applying to the context of hyper-parameter selection problem, each configuration of hyper-parameters can be considered as an expert. One round of horse-racing is equivalent to ten iterations of the AwPCSD algorithm. The loss after each round is computed as the RMSE error from the prediction of the next projection. We are in search for the configuration of hyper-parameters that returns the lowest cumulative error after the maximum number of iteration is reached, i.e. the one with the highest weight. More detail of the algorithm proposed in this work is explained in the next section.

7.1.2 Hedging hyper-parameter selection for CT reconstruction using TV regularised algorithm

The Hedging hyper-parameter selection in this work is an adaptation of the method described in [26] for the particular case of the Least Absolute Selection and Shrinkage Operator (LASSO) estimator, which is a popular estimator in high dimensional statistics. Apart from the LASSO

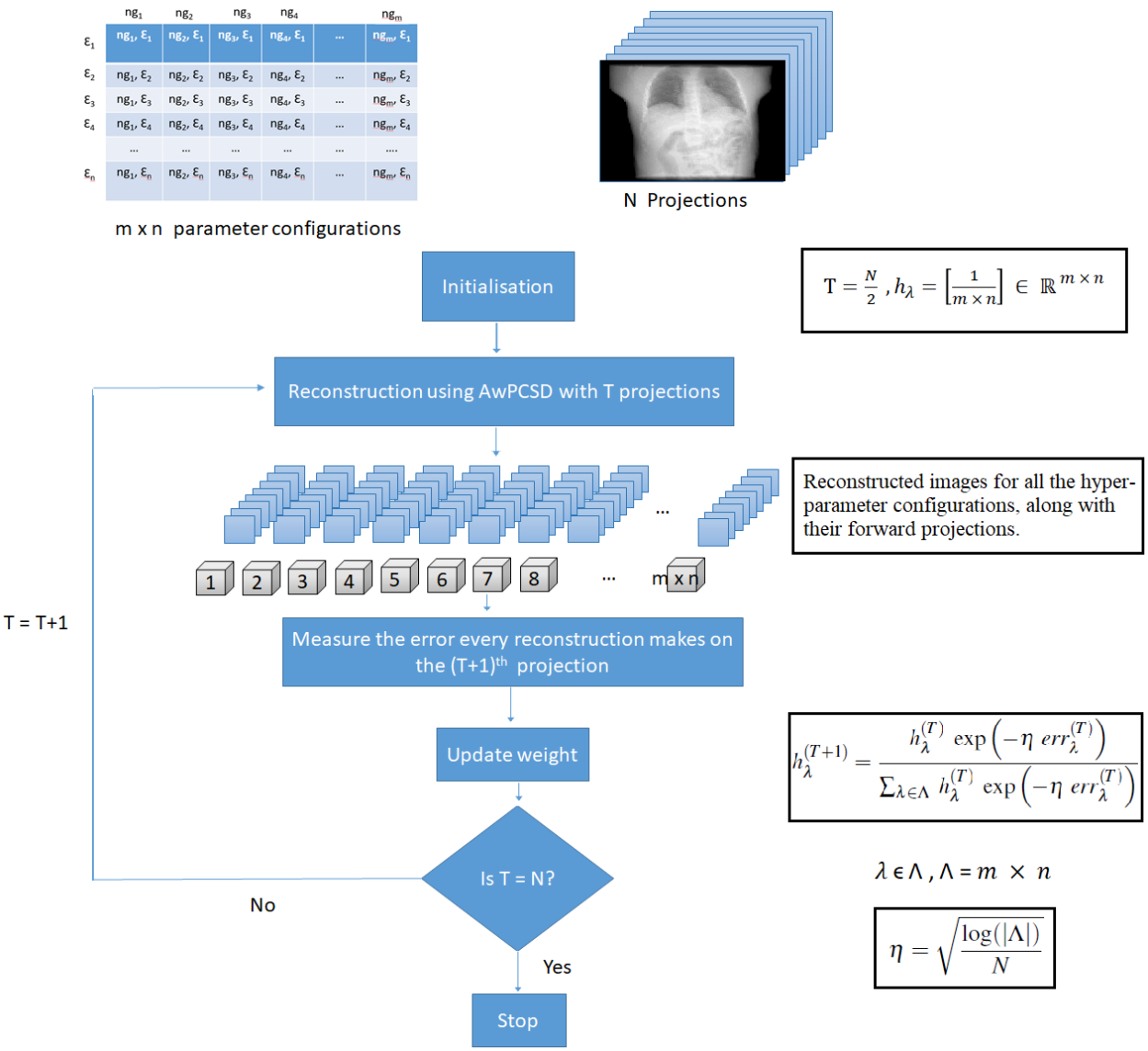


Figure 7-1: The diagram of hyper-parameter selection method using Freund and Shapire's hedge algorithm

estimator, the method can also be extended to other models such as two-stage estimation [25], [145], graphical models [89] and Robust PCA [16], etc.

The main idea behind the algorithm in this chapter is to select the hyper-parameters based on online learning, i.e. by adding one projection at each iteration after an initial starting image has been found using a pre-specified number of preliminary projections. The algorithm combines the Hedge method of Freund and Shapire with the AwPCSD reconstruction algorithm. The overall picture of the proposed algorithm is shown in figure 7-1.

For the implementation of AwPCSD algorithm, five hyper-parameters are required. The values of the following three hyper-parameters are fixed, following the empirical conclusion from our work proposed in [81] : Relaxation parameter (β) = 1, Reduction factor of relaxation parameter (β_{red}) = 0.99, Scale factor for adaptive-weighted TV norm (δ) = the 90th percentile of

the histogram of the reconstructed image from the OS-SART algorithm.

The remaining two hyper-parameters, TV sub-iteration number (ng) and data-inconsistency-tolerance parameter (ε), are the ones being calibrated using the Hedge algorithm in this work.

Let us now describe the hyper-parameter selection procedure in the context of this work. Firstly, N denotes a number of the available projection data. A finite set of hyper-parameter configurations (Λ) is selected, from which the value of each hyper-parameter configuration (λ) will be compared. According to the diagram in figure 7-1, number of $\Lambda = m \times n$, where $\lambda = (ng_m, \varepsilon_n)$ and $\lambda \in \Lambda$. Each value of hyper-parameter configuration (λ) can be interpreted as an 'expert'. All of these experts can be compared based on how well they predict the value of the next projection of data.

To set up all the hyper-parameter configurations, the range of candidates for each hyper-parameter is specified, i.e. TV sub-iteration number varies from ng_1 to ng_m and data-inconsistency-tolerance parameter varies from ε_1 to ε_n. Each hyper-parameter candidate is combined one-by-one to form $m \times n$ hyper-parameter configurations.

In the initialisation stage, a probability mass (h_λ) of all the hyper-parameter configurations is set to $\frac{1}{m \times n}$, to represent equal probability for all the possible candidates. The initial starting images are then reconstructed by AwPCSD algorithm using a pre-specified number of preliminary projections (T), which is initially set to half of the total number of projections, i.e. $T = \frac{N}{2}$. The method simply consists in running 10 iterations of the AwPCSD algorithm in parallel, each one with a different candidate value of the hyper-parameters $\lambda \in \Lambda$. Once the initial images are obtained, the Hedge algorithm then takes place in the following steps:

- The forward projections of the images reconstructed from all the hyper-parameter configurations are computed.
- For each hyper-parameter configuration, a prediction is performed for the next observed projection ($T+1$) and an RMSE error (err_λ) between the reference projection and the computed projection is calculated. The comparison is done in the projection domain, meaning that the training is possible from real data (without the exact image from simulation data).
- The vector of errors for all the hyper-parameter configurations are used to update the probability mass (h_λ) according to the rule [6]:

$$h_\lambda^{(T+1)} = \frac{h_\lambda^{(T)} \exp\left(-\eta \; err_\lambda^{(T)}\right)}{\sum_{\lambda \in \Lambda} h_\lambda^{(T)} \exp\left(-\eta \; err_\lambda^{(T)}\right)}. \tag{7.1}$$

The value of η is defined as following:

$$\eta = \sqrt{\frac{\log(|\Lambda|)}{N}}. \tag{7.2}$$

After the probability mass (h_λ) is updated, the hyper-parameter configurations which produce the probability of less than 10% of the highest probability are discarded to speed up the

process. Then, a new projection is added to the set of projection used to reconstruct an image. This newly added projection is the next observed projection, i.e. the projection number $(T+1)$. The whole method is then repeated again until the number of T reaches the number of available projection data N.

At the end of the algorithm, after n steps of the algorithm is implemented, the probability mass for each hyper-parameter configuration (h_λ) is plotted. This probability vector can be used for the purpose of predictor aggregation. The best hyper-parameter configuration is considered as the one with the highest probability mass.

In machine learning and computer science literature, the Hedge algorithm has been extensively studied according to the survey paper [6]. The main result of the Hedge algorithm is that if the prediction errors ($err_\lambda^{(i)}$) are in $(0,1)$, or are at least bounded, then choosing the parameter configuration (λ) using the probability mass $(h^{(N)})$ after n steps provides a prediction error which is almost as accurate as the prediction error incurred by the best predictor. This is equivalent to the reconstruction algorithm governed by the best value of the parameter configuration (λ) [46], which is an unknown set and can only be obtained from a number of trials and errors.

7.2 Results

7.2.1 Digital XCAT phantom

Some numerical studies are implemented to evaluate the Hedge hyper-parameter selection algorithm. Initially, the digital XCAT phantom [115] is used as a data set. In the same way as the experiments in the previous chapters, Poisson and Gaussian noise is added to the input projection data to simulate a realistic noise.

In the first study, 50 projections from the data set are taken as an input. These projections are equally sampled over 360°. The AwPCSD reconstruction algorithm requires 5 hyper-parameters. The values of all hyper-parameter configurations are shown in the following table.

Table 7.1: Values of each hyper-parameter configuration for this study

Hyper-parameter	Values
Data-inconsistency-tolerance parameter (ε)	0,50,70,100,200,500,2×10^3,1×10^4,1×10^5,5×10^5
TV sub-iteration number (ng)	2,4,6,8,10,12,14,16,18,20,22,24,26,28,30
Relaxation parameter (β)	1
Reduction factor of relaxation parameter (β_{red})	0.99
Scale factor for adaptive-weighted TV norm (δ)	0.0212

In the same way as the experiments in the previous chapter, the three hyper-parameters (β, β_{red} and δ) are specified the fixed values. Also, the same values of hyper-parameters are

specified, in order to compare the results with the ACO algorithm. There are 10 values of ε and 15 values of ng hyper-parameters. The combination of these values leads to 150 different hyper-parameter configurations.

As the total number of projection data (N) in this study is 50, the initial starting image for the Hedge algorithm is reconstructed using 25 projections (T=25). Then, T is incremented by 1 for the next iteration. The algorithm is iteratively implemented until T reaches N. The probability mass of all hyper-parameter configuration at the end of the implementation is plotted in figure 7-2.

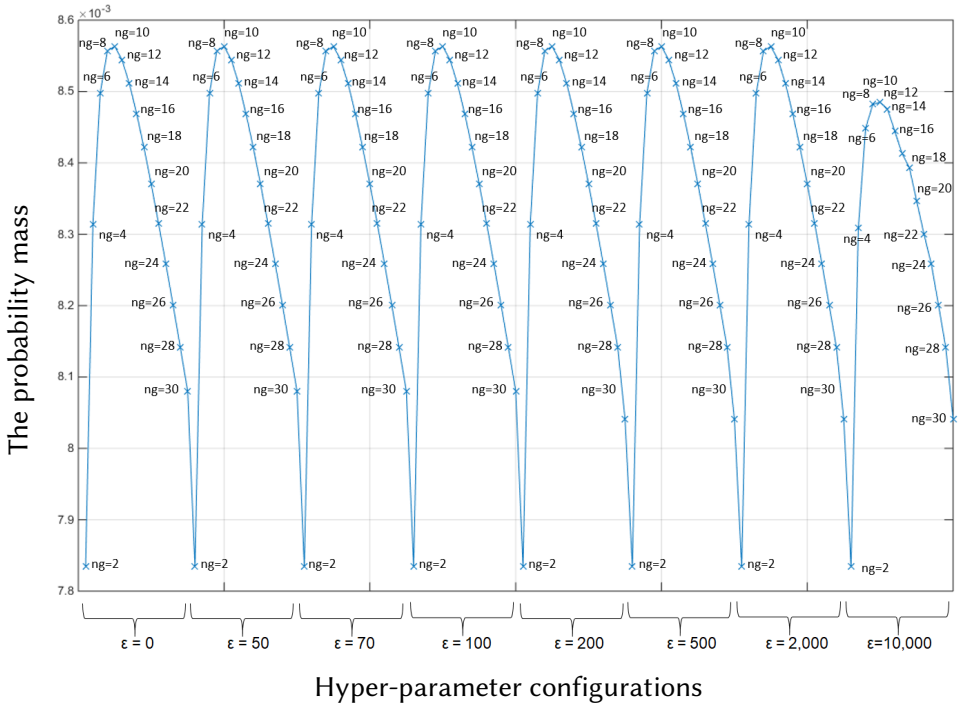
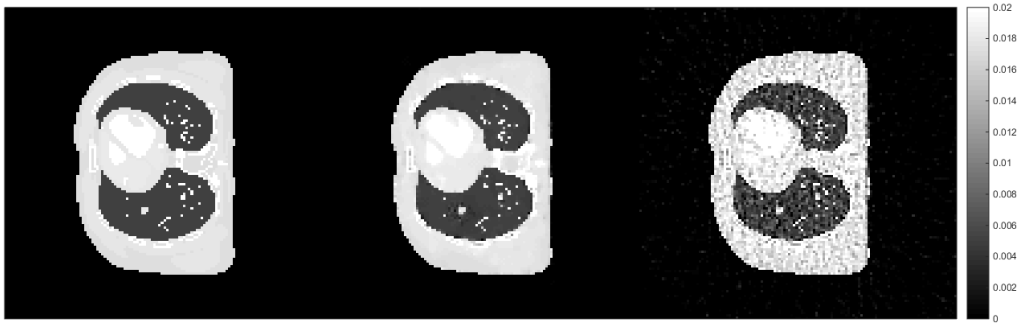


Figure 7-2: The probability mass of all the hyper-parameter configurations after 25 iterations of the Hedge algorithm (T = 25 to 50)

Since the algorithm had discarded the hyper-parameter configurations which produce the probability of less than 10% of the highest probability, the number of remaining hyper-parameter configurations after this experiment is 120 configurations, as displayed in figure 7-2. The best hyper-parameter configuration is the one with the highest probability and vice versa. According to figure 7-2, the highest probability is the configuration number 5 ($ng = 10, \varepsilon = 0$). The lowest probability is the configuration number 16 ($ng = 2, \varepsilon = 50$). The probability plot in figure 7-2 is presented per hyper-parameter combination consisting of two values (ng and ε) since it reflects the performance of both hyper-parameters, simultaneously. The combinations of all the initial values of hyper-parameters are defined at the start of the algorithm. Both ng and ε values are combined in a periodic manner to create 150 different combinations of hyper-parameters before being implemented the Hedge algorithm. Each value of hyper-parameters is labelled in the plot to give an information of each hyper-parameter combination. Cross-sectional slices of the reconstructed images from the best and the worst hyper-parameter configurations are shown

in figure 7-3.



(a) (b) (c)

Figure 7-3: The cross-sectional slices of the reconstructed image from: (a) the exact phantom, (b) the best hyper-parameter configuration, and (c) the worst hyper-parameter configuration. The display window is [0-0.02].

It is obviously seen in figure 7-3 that the reconstructed image from the best configuration of hyper-parameters is better to recover the important features of the image and looks rather similar to the exact phantom image. On the other hand, the result from the worst configuration of hyper-parameters suffers from noise and much more different from the exact phantom from the visual inspection.

7.2.2 Performance evaluation

In order to evaluate the performance of the hyper-parameter selection algorithm using the hedge method, the result is compared with the previously implemented algorithms in the previous chapter, i.e. the cross-validation and the ACO algorithms. In this section, the results of the XCAT phantom of the 3 methods are compared. Starting with the same initial ranges of the hyper-parameters as displayed in the table 7.1, the best hyper-parameter configurations as found by three methods are summarised in table 7.2.

Table 7.2: Best hyper-parameter configurations as found by the hedge, the ACO and the cross-validation algorithms.

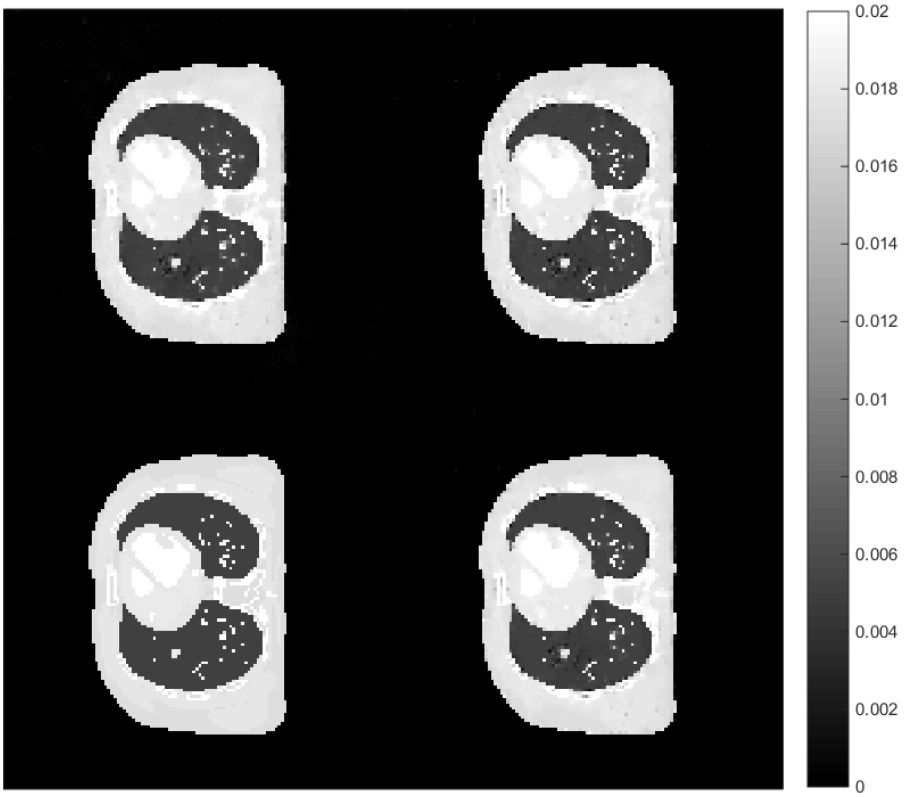
Algorithms	TV sub-iteration number (ng)	Data-inconsistency-tolerance parameter (ε)
The hedge algorithm	10	0
The ACO algorithm	10	2,000
The cross-validation algorithm	8	0

It is worth mentioning here that the $\varepsilon = 0$ for the hedge and cross-validation algorithms does not mean that the algorithms were able to achieve no error between the predicted and observed

projection data. In this case, what happened was that the algorithms aimed to reach the zero error point, when the hyper-parameter configuration under study contains $\varepsilon = 0$, but was forced to stop due to the maximum number of iteration was reached first. Thus, the hyper-parameter configurations which contain $\varepsilon = 0$ was evaluated based on this attempt, in combination with the performance as affected by the other hyper-parameters.

In order to compare the results visually, the cross-sectional slices of the reconstructed images from the AwPCSD algorithm using best hyper-parameter configurations of 3 methods are shown in figure 7-4.

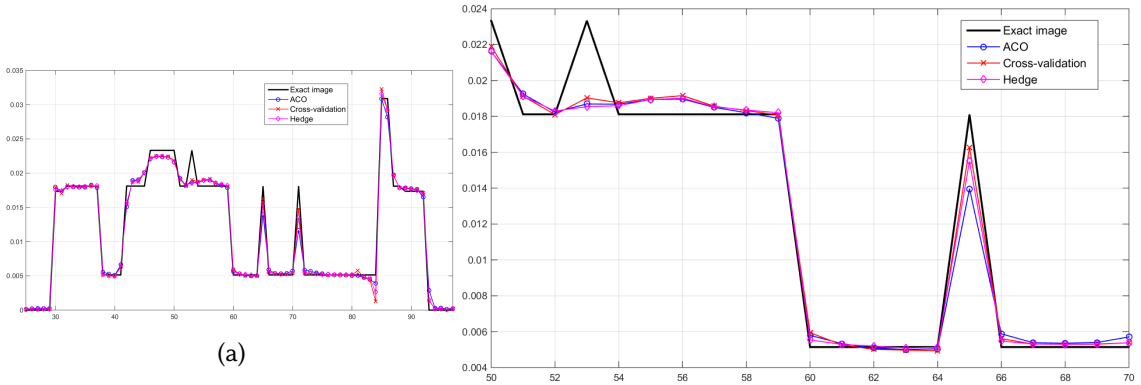
(a) ACO algorithm (b) Cross-validation algorithm



(c) Exact image (d) Hedge algorithm

Figure 7-4: The cross-sectional slices of the reconstructed images from 3 methods of hyper-parameter selection algorithms, in comparison with the exact image. The display window is [0-0.02].

From visual inspection, the results of 3 algorithms are very similar to each other. There is no outstanding differences that could be observed from the cross-sectional slices of the reconstructed images. The results are analysed further by plotting one-dimensional profile of each reconstructed images along one arbitrary row of the images. The one-dimensional profiles plots are shown in figure 7-5.



(a) (b)

Figure 7-5: One dimensional profile plots of the reconstructed images from three hyper-parameter selection methods along one arbitrary row of the images, in comparison with the exact image: 7-5a plots along pixel numbers 25 to 97, 7-5b zoom in along pixel numbers 50 to 70.

The one-dimensional profile plots also show the similarity of results from the three algorithms. Taking a closer look in figure 7-5b, it can be seen that the profile of the cross-validation method is closer to that of the exact image in some areas such as pixel numbers 53 and 65. However, the result from the hedge algorithm is more aligned with the exact image between pixel numbers 60-64 and 66-70.

In order to quantify the difference between the results, the Universal Quality Index (UQI) and the relative 2-norm errors, as defined in equations 6.4 and 6.5, are computed. Table 7.3 displays the relative errors, UQI values and the computational time taken to implement the hyper-parameter selection for all 3 algorithms. The computational time is measured using the same testing computer (Intel Core i7-4930K CPU @3.40GHz with 32 GB RAM and GPU: NVDIA GeForce GT 610).

Table 7.3: Relative errors, UQI and computational time of image reconstruction results from 3 hyper-parameter selection algorithms. (Boldface numbers indicate the best result)

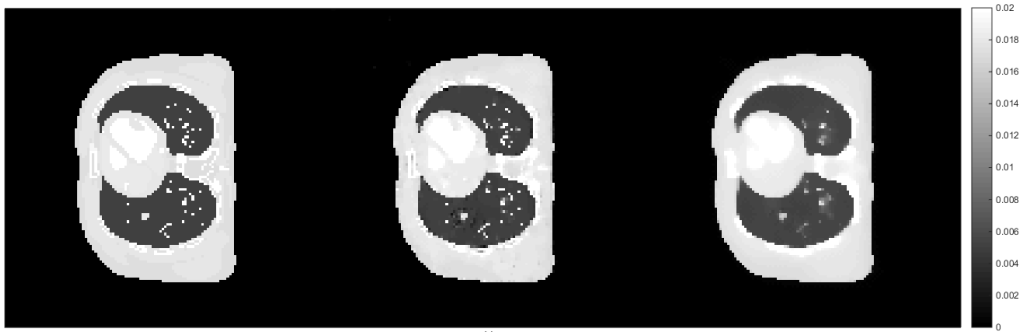
Algorithms	Relative errors (%)	UQI	Computational time (hours)
The hedge algorithm	**4.9349**	**0.9972**	16.12
The ACO algorithm	6.6979	0.9959	**1.45**
The cross-validation algorithm	5.2597	0.9970	47.15

Quantitatively, the result from the hedge algorithm is the best among the 3 algorithms with the lowest relative error and the highest UQI. When comparing the computational time taken to finish each algorithm, the ACO algorithm is the one with the shortest time, $\approx$ 1.45 hours. The computational time of the cross-validation algorithm is the longest, $\approx$ 47.15 hours. The hedge algorithm is the middle ground between the two algorithm, $\approx$ 16.12 hours.

Further improvements can be implemented in an attempt to reduce the computational time of the Hedge algorithm. The first improvement is to discard more hyper-parameter configurations from one iteration to another. In the first attempt, the configurations which produce the probability of less than 10% of the highest probability are discarded before proceeding to the next iteration. This threshold can be reduced to 5% to only let the configurations with the best probability pass through. The second improvement is to re-select the pre-specified number of preliminary projections (T). In the first attempt, this number is specified as $T = \frac{N}{2} = 25$. We can try to set T to $\approx$ 30-40 and the computational time should be reduced further. Next section presents the experiments to improvement the computational time of the Hedge algorithm.

Further improvements of the hedge hyper-parameter selection algorithms

• The first improvement of the algorithm is implemented by discarding the hyper-parameter configurations which produce the probability of less than 5% of the highest probability before proceeding to the next iteration. The other settings remain the same. At the end of the implementation, there are 104 hyper-parameter configurations remaining from the starting 150 configuration. The computational time is improved from 16.12 to **14.45 hours**. In term of the result, the best hyper-parameter configuration is still similar to the base experiment, i.e. $ng = 10$ and $\varepsilon = 0$. However, the worst configuration has changed to $ng = 28$ and $\varepsilon = 0$, which produces the improved result from previous experiment since more low-performance configurations are discarded during the process. The reconstructed images from the first improvement is shown in figure 7-6.

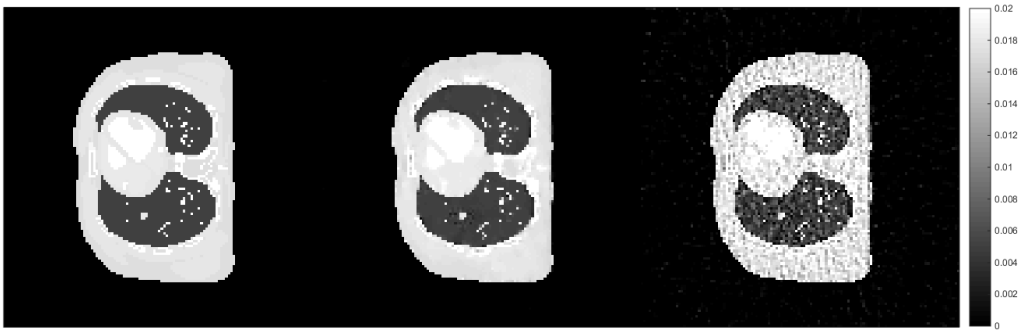


(a) (b) (c)

Figure 7-6: The cross-sectional slices of the reconstructed image from the first improvement: (a) the exact phantom, (b) the best hyper-parameter configuration, and (c) the worst hyper-parameter configuration. The display window is [0-0.02].

• The second improvement is implemented by specifying $T = 35$, instead of $T = 25$, at the beginning of the algorithm. This means that the number of projections used to reconstruct a preliminary image is 35 projection views, which will make the algorithm start with a better quality of preliminary image. However, the algorithm will only be implemented for 15 iterations (from $T = 35$ to 50). The other setting remain the same. At the end of the implementation, there are 120 hyper-parameter configurations remaining from the starting 150 configurations.

The computational time is drastically improved from 16.12 to **10.50 hours** with similar result for the best hyper-parameter configuration, i.e. $ng = 10$ and $\varepsilon = 0$. The worst hyper-parameter configuration from this experiment is $ng = 2$ and $\varepsilon = 0$. The cross-sectional slices are shown in figure 7-7.



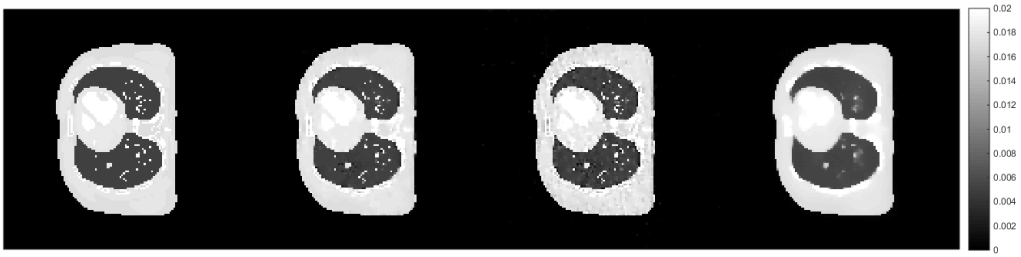
(a) (b) (c)

Figure 7-7: The cross-sectional slices of the reconstructed image from the second improvement: (a) the exact phantom, (b) the best hyper-parameter configuration, and (c) the worst hyper-parameter configuration. The display window is [0-0.02].

• The third improvement combines the first and the second improvements by specifying $T = 40$ and discarding the hyper-parameter configurations which produce the probability of less than 5% of the highest probability before proceeding to the next iteration. In this case, the algorithm will be implemented for 10 iterations (from $T = 40$ to 50). At the end of implementation, the same best and worst hyper-parameters as the second improvement are obtained (and hence the same cross-sectional slices of the results as shown in figure 7-7) with the computational time reduced to $\approx$ **7.36 hours**.

Up to this point, it can be concluded that by changing some settings from the base experiment such as T from 25 to 40 and the threshold to discard the hyper-parameters from 10% to 5% as in the third experiment resulted in the same set of best hyper-parameter configuration with drastic reduction of computational time of the training (from $\approx$16.12 to 7.36 hours). Next step, we try to see further possible alternative to reduce the computational time without violating the quality of the result. In the base experiment, 10 iterations of the AwPCSD reconstruction is implemented per one step of the Hedge algorithm. This number is reduced to 5 iterations of the AwPCSD algorithm in one step of the Hedge algorithm and the following experiments are performed.

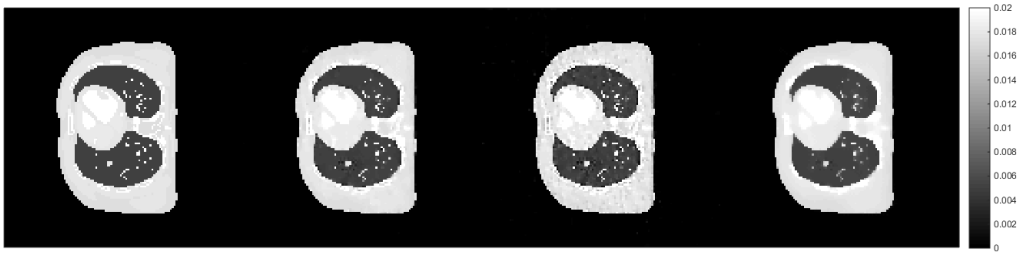
In the first experiment, T is specified as 40 and the hyper-parameter configurations which produce the probability of less than 5% of the highest probability are discarded. At the end of the implementation, there are 120 configurations remaining with the best and worst hyper-parameter configurations being ($ng = 6$,$\varepsilon = 0$) and ($ng = 30$,$\varepsilon = 0$), respectively. The computation time taken to complete the implementation is **4.25 hours** and the cross-sectional slices of the results are compared with the result from the third improvement as shown in figure 7-8.



(a) (b) (c) (d)

Figure 7-8: The cross-sectional slices of the reconstructed image from:(a) the exact phantom, (b) the best hyper-parameter configuration from the third improvement, (c) the best hyper-parameter configuration from the first experiment and (c) the worst hyper-parameter configuration from the first experiment. The display window is [0-0.02].

It can be observed that the reconstructed image from the best configuration of the first experiment (T=40, 5 iterations, threshold = 5%) in figure 7-8(c) is slightly inferior in the image quality to the one from the third improvement improvement (T=40, 10 iterations, threshold = 5%) in figure 7-8(b). The second experiment is then set up to observe what happen if T is changed to 25 instead of 40 while the other settings remain the same (5 iterations, threshold = 5%). The cross-sectional slices of the results are compared with the result from the third improvement as shown in figure 7-9.



(a) (b) (c) (d)

Figure 7-9: The cross-sectional slices of the reconstructed image from:(a) the exact phantom, (b) the best hyper-parameter configuration from the third improvement, (c) the best hyper-parameter configuration from the second experiment and (c) the worst hyper-parameter configuration from the second experiment. The display window is [0-0.02].

At the end of the second experiment, 80 configurations are remaining with the best and worst hyper-parameter configurations being ($ng = 6, \varepsilon = 0$) and ($ng = 20, \varepsilon = 0$), respectively. The computation time is **7.42 hours**, which is longer than the first experiment since the algorithm performed more steps (from T = 25 to 50). It can be noted that the only difference in the result from the first experiment is the image of the worst hyper-parameter configuration.

Relatively speaking, the two experiments returned the same best configuration of hyper-parameters, i.e. $ng = 6$ and $\varepsilon = 0$. The reconstructed image using this set of hyper-parameters, however, is not as good a quality of image as the reconstructed image using the set of hyper-parameters obtained from the experiment setting of the third improvement, i.e. $ng = 10$ and

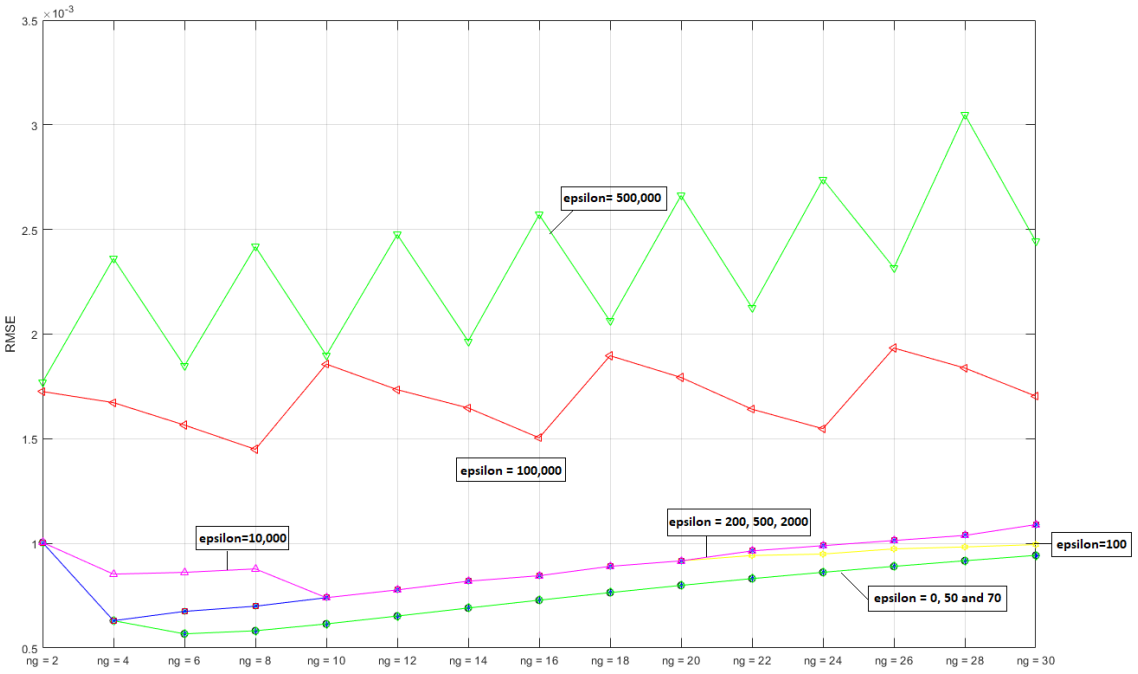


Figure 7-10: Plots of the RMSE errors computed from the reconstructed images using the AwPCSD algorithm with each value of hyper-parameters presented in table 7.1. Y axis presents the RMSE and X axis presents different TV sub-iteration numbers (ng). The values of ε are annotated for each plot in the figure.

$\varepsilon = 0$. These two experiments showed that reducing the number of AwPCSD iteration per step of the Hedge algorithm sacrifices for the inferior quality of the images reconstructed from each hyper-parameter configuration, which are used for a comparison during the process. Eventually, at the end of the implementation, this results in the algorithm returning the hyper-parameter configuration that produces not as good quality of image as the other configurations as chosen from the Hedge algorithm with more iterations of the AwPCSD algorithm in one step.

It can be concluded that the best way to implement the Hedge algorithm for the XCAT phantom data is to set the experimental setting to be in line with the third improvement, i.e. specifying 10 iterations of the AwPCSD per step, $T = 40$ and discarding the hyper-parameter configurations which produce the probability of less than 5% of the highest probability before proceeding to the next iteration. With this setting, the same best set of hyper-parameters for the given data is achieved at the computational time of 7.36 hours, which has improved drastically from the best experiment of 16.12 hours.

Up to this point, we have seen the results from the 2 proposed hyper-parameter selection algorithms, the ACO and Hedge algorithms, and the cross-validation algorithm. Lastly, an experiment is designed to reconstruct images using the AwPCSD algorithm with all values of hyper-parameters, as shown in table 7.1, one by one. The XCAT phantom data with the same experimental settings as previous experiments is used. Root mean squared error (RMSE) is measured from each reconstructed image using each value of hyper-parameters and presented in figure 7-10.

From figure 7-10, we can see that the majority of RMSE values increase with increasing values of ε. There are some groups of ε that return the same RMSE values across the entire range of varying ng such as the groups of $\varepsilon = 0, 50, 70$ and $\varepsilon = 200, 500, 2,000$. These are because the algorithm returns the same output images with these ranges of specified values of ε due to several factors such as the stopping criterion and the maximum number of iteration. The final residual norm $||Ax - b||_2$ of the reconstructed image does not always be the same as the input ε specified to the algorithm. The detailed discussion of this issue is presented in section of data-inconsistency-tolerance parameter in chapter 5.

The values of hyper-parameters which contribute to the reconstructed image with the lowest RMSE in this experiment are $\varepsilon = 0, 50, 70$ and $ng = 6$. These values are slightly different to the best ones found by the three methods, presented in table 7.2 earlier. The best hyper-parameter configurations found by the Hedge, ACO and the cross-validation algorithms are $\varepsilon = 0 \& ng = 10$,$\varepsilon = 2,000 \& ng = 10$ and $\varepsilon = 0 \& ng = 8$, respectively. Although the values of hyper-parameters found from the 2 proposed algorithms do not yield the image with the lowest RMSE error, as confirmed by the results of this experiment, the proposed algorithms are fully automated and do not require any actions from the user between the implementation. The 2 hyper-parameter selection algorithms proposed in this thesis can be further improved to address all the drawbacks and shorten the computational time, which will make them useful for the future use.

Experiments with different number of available projections (N)

All previous experiments in this chapter are based on the total number of projection data equals to 50 ($N = 50$). In this section, experiments with different number of available projections are conducted to observe how the hyper-parameter selection algorithm works in these situations.

In the first experiment, 100 projection images ($N = 100$) are used as an input to the Hedge algorithm. A pre-specified number of preliminary projection (T) is set to $\frac{N}{2} = 50$ in this case. The probability mass of all the hyper-parameter configurations at the end of the implementation is plotted in figure 7-11.

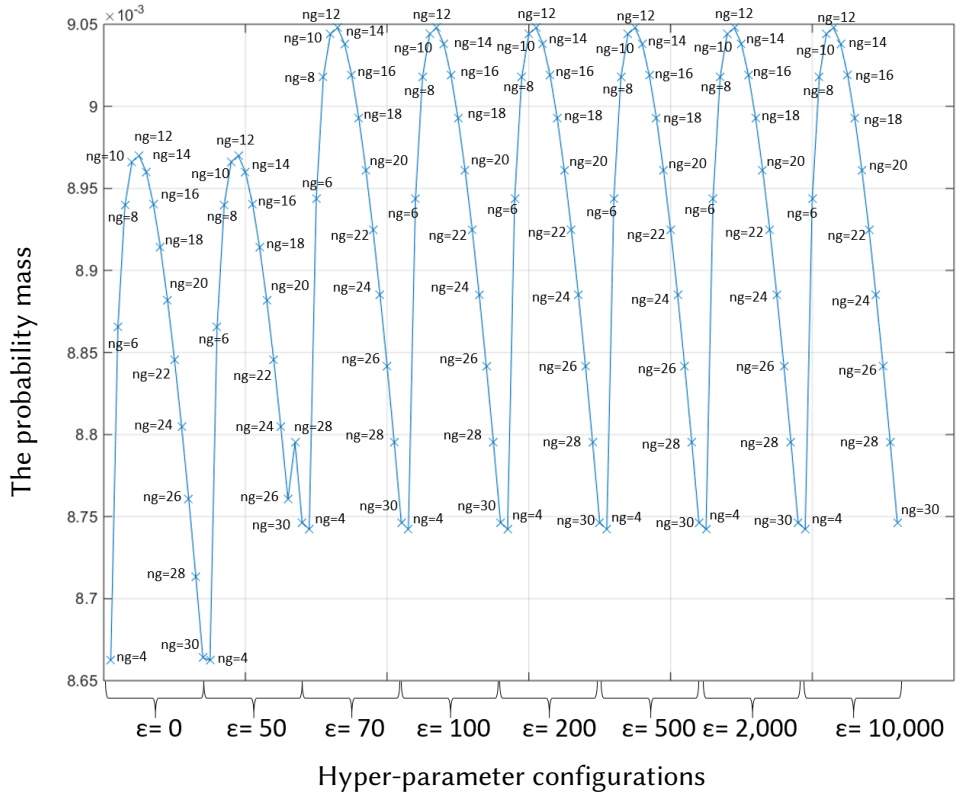


Figure 7-11: The probability mass of all the hyper-parameter configurations from the experiment with $N = 100$ and $T = 50$.

The remaining number of hyper-parameter configurations after the experiment is 112 configurations. The best hyper-parameter configurations with the highest probability is $(ng = 12, \varepsilon = 70)$. The cross-sectional slice of reconstructed image with the best hyper-parameter configuration found in this experiment is displayed in figure 7-12.

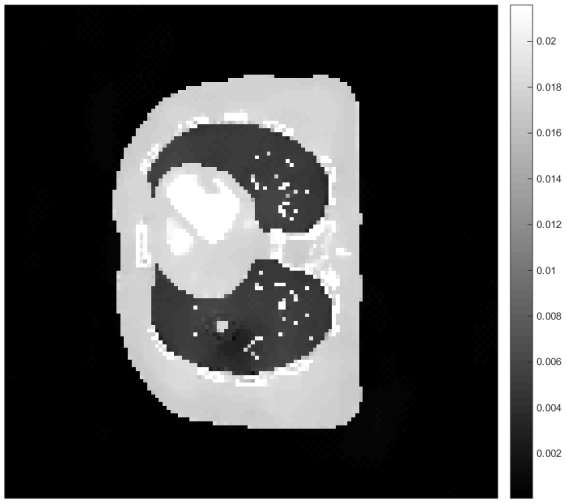


Figure 7-12: The cross-sectional slices of the reconstructed image from the best hyper-parameter configuration from the experiment with $N = 100$ and $T = 50$.

Regarding the computational time, the experiment with $N = 100$ and $T = 50$ takes approximately 3 days to finish. The time is much longer compared to previous experiments with $N = 50$ (≈ 7.36 hours).

Next, 40 projection images ($N = 40$) are used as an input to the Hedge algorithm. In this case, T is specified as 30. The probability mass of all the hyper-parameter configurations at the end of the implementation is plotted in figure 7-13.

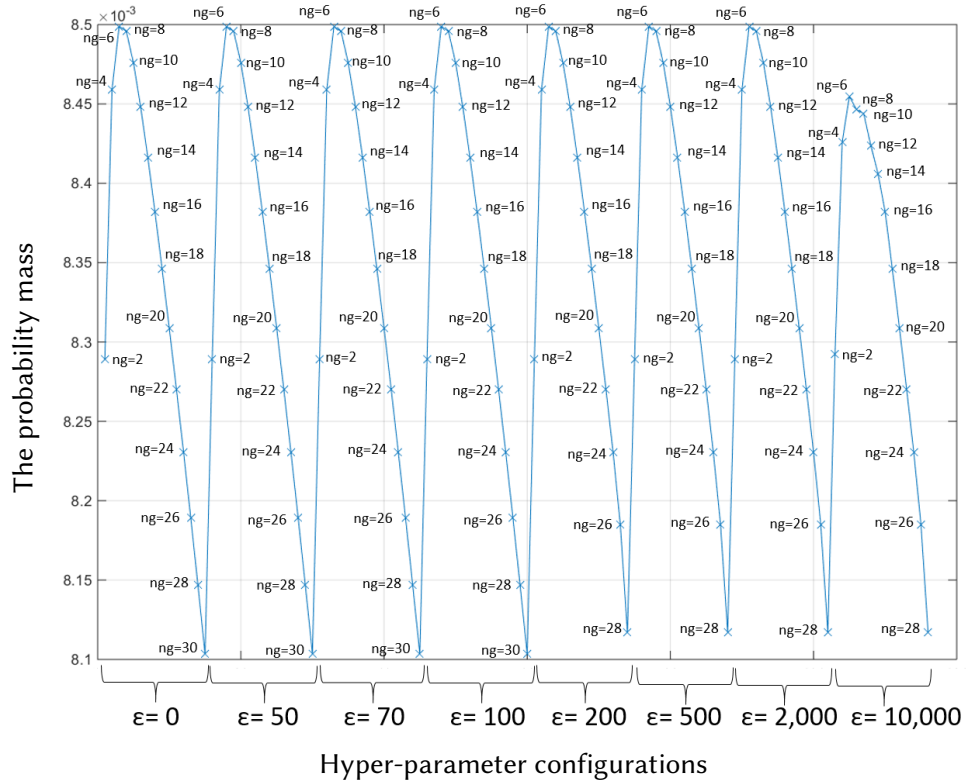


Figure 7-13: The probability mass of all the hyper-parameter configurations from the experiment with $N = 40$ and $T = 30$.

The remaining number of hyper-parameter configurations after the experiment is 116 configurations. The best hyper-parameter configurations with the highest probability is $(ng = 6, \varepsilon = 70)$. The cross-sectional slice of reconstructed image with the best hyper-parameter configuration found in this experiment is displayed in figure 7-14. The computational time taken for the experiment with $N = 40$ and $T = 30$ is $\approx$7 hours 23 minutes.

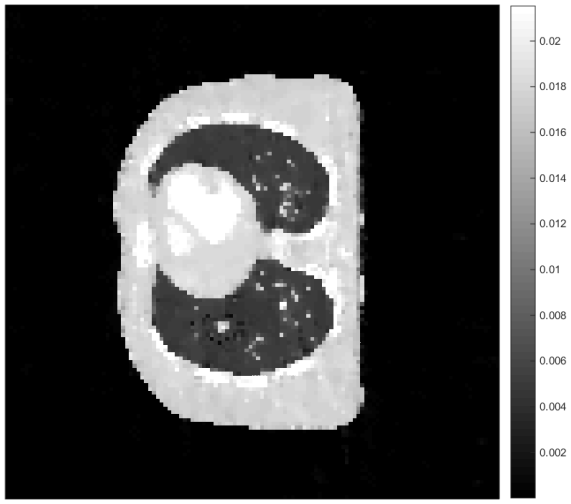


Figure 7-14: The cross-sectional slices of the reconstructed image from the best hyper-parameter configuration from the experiment with $N = 40$ and $T = 30$.

According to the experimental results in this section, several conclusions can be made. Firstly, both the number of projection data (N) and a pre-specified number of preliminary projection (T) have strong effects on the computational time of the algorithm. With higher N, T will also be higher accordingly since the algorithm has more projections to work on. The initial value of T, whether it is set at $T = \frac{N}{2}$ or higher would affect how many new projections to be added during the prediction process of the hyper-parameter selection algorithm. We have already shown the results with different numbers of T for a given number of N in the earlier experiments.

In this section, with 2 different numbers of N chosen in the experiments, i.e. $N = 100$ and $N = 40$, we can see that the computational time for $N = 100$ is much longer than those of $N = 40$ and $N = 50$. This is because for each iteration of hyper-parameter selection algorithm, there are more projection for the reconstruction with each hyper-parameter. Hence, more time required to finish the process.

The total number of projection data (N) is directly related to the amount of dose delivered to patients, as discussed in details in section 5.4 of chapter 5. However, the role of N in the context in this chapter is served as an input to the hyper-parameter selection algorithm. The aim is to find an efficient way to select hyper-parameters which would be most suited to a given set of data and produce reconstructed image with good quality.

7.3 Discussion

It is obviously seen from the experiments in this chapter that several factors can affect the computational time as well as the quality of the reconstruction result. First factor, T specifies how many projections to be used for a preliminary image, which will also relate to the number of steps for the Hedge algorithm. For instance, when T is specified as 25, the algorithm will only have 25 projections to reconstruct the preliminary image and the algorithm will be implemented for 25 steps until all the projections are incorporated. If T is set to 40, it will be 40 projections for

the reconstruction of the preliminary image and 10 steps for the algorithm to be implemented. According to the experiments, specifying T as 40 return the same result as T=25 with much reduced computational time can be implied that more number of projections for the preliminary image can be compensated for the fewer number of steps for the algorithm.

The second factor is the threshold to discard the hyper-parameter configurations. Discarding more low-performance configurations before proceeding to the next step of algorithm could help to reduce the computational time and does not affect the result since we are more interested in the configuration with high probability distribution. The last factor is the number of iteration of AwPCSD algorithm per one step of the Hedge algorithm. This can be obviously seen from the final two experiments that reducing the number of AwPCSD iteration per step sacrifices for the inferior quality of the images reconstructed from each hyper-parameter configuration, which eventually make the algorithm chooses the hyper-parameter configuration that produces not as good quality of image as the other configurations chosen from the Hedge algorithm with more iterations of the AwPCSD algorithm in one step.

7.4 Summary

Taking the motivation from the previous chapter, this chapter considers the hyper-parameter selection problem from a new perspective by combining the Hedge method of Freund and Shapire with the AwPCSD reconstruction algorithm. The Hedge method is the online prediction model, which is commonly used to solve dynamic allocation problems. In this context, the method is adopted to select the best hyper-parameters for the TV regularised reconstruction algorithm by considering the hyper-parameter selection problem as the experts problem. Each configuration of hyper-parameters is assumed to be 'an expert' and its advice is the corresponding reconstruction using each configuration. In one step of the Hedge algorithm, several instances of the AwPCSD algorithm are implemented by using each hyper-parameter configuration. After each step, the loss of each advice is computed as the RMSE error from the prediction of the next projection. A probability distribution of all the experts are updated based on the computed errors. Then, new projection is incorporated for the next step of the algorithm so that a prediction model learns as it goes through the experience when more aspect of the problems are observed. The process is repeated until all the available projection is incorporated into the reconstruction. One expert's advice will be chosen by the user as the best advice based on how well they predict the results throughout the implementation. This means that we are in search for the configuration of hyper-parameters that returns the lowest cumulative error after the maximum number of iteration is reached, i.e. the configuration of hyper-parameters with the highest probability mass.

The Hedge algorithm is evaluated by using 50 projection views equally sampled over 360° digital XCAT phantom. In the same way as the previous chapter, the same initial values of ng and ε hyper-parameters are specified so that the results can be compared with the ACO algorithm. Starting with the base experiment, the number of projections used to reconstruct a preliminary image (T) is specified as half of the number of available projections, i.e. $T = \frac{50}{2} = 25$. Also, the

threshold to discard the hyper-parameter configurations before proceeding to the next step is set to 10%, meaning that any configurations which produce probability of less than 10% of the highest probability will be discarded.

After the implementation, the reconstruction result from the best hyper-parameter configuration obtained from the Hedge algorithm ($ng = 10$, $\varepsilon = 0$) is compared with the ones from the ACO ($ng = 10$, $\varepsilon = 2,000$) and the cross-validation ($ng = 8$, $\varepsilon = 0$) algorithms. The reconstructed images from 3 algorithms are very similar to each other as well as the exact phantom image from visual inspection and one-dimensional profile plots. This means that AwPCSD reconstruction using the sets of hyper-parameters obtained from 3 algorithms is able to produce the images with as good quality as that of the exact phantom image. However, the results as measured from the quantitative metrics (relative error and the UQI) showed that the result from the Hedge algorithm is the best among the 3 algorithms with the lowest relative error and the highest UQI.

Another important factor to be considered apart from the quality of the reconstructed images is the computational time taken to complete the implementation. With the same setting, the cross-validation algorithm takes $\approx$ 47.15 hours, the Hedge algorithm takes 16.12 hours and the ACO algorithm takes 1.45 hours. Further improvements have been experimented in an attempt to reduce the computational time without violating the quality of the result and it can be concluded that the best setting to implement the Hedge algorithm is to specify 10 iterations of the AwPCSD per step, $T = 40$ and discarding the hyper-parameter configurations which produce the probability of less than 5% of the highest probability before proceeding to the next iteration. This setting results in the computational time of $\approx$7.36 hours with the same set of best hyper-parameter configuration as that of the base experiment.

Chapter 8

Conclusions and future work

The main objective of this thesis focuses on the reconstruction of CT imaging using limited number of projection data, in order to reduce the amount of radiation dose delivered to patient. This area of research has been active for over decades with numerous numbers of work developed to overcome the problems arising from the CT reconstruction with an insufficient number of data. These problems have been extensively studied in this thesis in different aspects as can be summarised in this chapter, together with the contributions this thesis have made.

Firstly, the acquisition of the CT projection is studied. The aim is to thoroughly understand how the CT projection is acquired, especially how the forward operators are computed. The forward operator is one of the important building block of the CT reconstruction algorithms. The accuracy of the reconstruction process depends highly on the forward operator. The system matrix A is computed based on the intersection of the X-ray and the image voxels. Even though the forward operators are computed based on the same principle, different geometries of the CT imaging require different approaches to calculate the element of system matrix A. The more dimensions, the more complicated the calculation gets, as well as the memory it takes to store the system matrix A. The chapter 2 of the thesis is devoted to explain how the forward operators of the 2D and 3D CT geometries are calculated using the MATLAB software, which later on is the based software for all the algorithms developed in this thesis. It is significant to understand the formation of the forward operator and come across the difficulty in storing the system matrix A, which is one of the complications of solving the CT reconstruction problem.

Then, the iterative algorithms are focussed due to their efficiency in reconstructing an image using limited number of data. The implementations as well as the advantages and disadvantages of the existing algorithms are studied to understand the basic principles and behaviour of the algorithms. The iterative algorithms work as a closed-loop system where a discrepancy between the projection of the estimated image and the measured data is compared for each iteration. This discrepancy is used to update the estimated image for the next iteration and so on, until the stopping criterion are met. The reconstructed images obtained from the standard iterative algorithms such as SART,SIRT,CGLS and OS-SART are promising. However, the improved quality of the results can be achieved by incorporating a prior knowledge about the structure of CT images to push the reconstruction algorithms towards a specific solution among multiple

possibilities. This thesis is interested in the regularisation algorithms that use the total variation (TV) norm of the image as an objective function to be minimised.

The common TV regularised algorithms are based on the two-stage approach, which consists of projection onto convex sets (POCS) to enforce the data fidelity and the positivity constraints and the steepest descent in the TV minimisation stage. The goal of the algorithms is to provide the image with minimum TV that satisfies a given error tolerance. The TV regularised algorithms have been very successful in the past decade. However, there are 2 main concerns of the implementation of TV regularised algorithms which shape the development of the algorithms in a number of ways. The first concern is the over-smoothing of the reconstructed image due to the assumption of piecewise constant of the reconstructed image. The over-smoothing is commonly occurred around the edges of the image, which leads to the loss of some important structural information of the image, especially for the medical imaging. The second concern is the selection of hyper-parameters to control the effects between the constraints and objective function of the TV regularised algorithms.

Given the background knowledge and research gaps, the following contributions have been made throughout this thesis:

The adaptive-weighted projection-controlled steepest descent (AwPCSD) reconstruction algorithm is proposed. It is the TV regularised algorithm, which is developed based on the existing algorithm, the projection-controlled steepest descent (PCSD), by replacing the conventional TV norm in the steepest descent step with the adaptive-weighted TV norm. The goal of the AwPCSD algorithm is to improve the edge preservation of the reconstructed image with fewer numbers of sensitive hyper-parameters to be tuned comparing to the ASD-POCS algorithm.

In the adaptive-weighted TV norm, the anisotropic edge property of an image is considered by adding a weight for each dimension to the conventional TV norm, in order to take the change in local voxel intensities into consideration. In a non-edge region of an image, which normally have a smaller change in local voxel intensity, a stronger weight maybe given to emphasise the TV minimisation. On the other hand, when the change is larger in the edge region, a weaker weight shall be given to alleviate the TV minimisation and preserve the edges of the reconstructed image. This weighting process allows better control over the effects that the TV minimisation process have for different gradient of the image. With this feature of the AwPCSD algorithm, the problem of over-smoothing of the reconstructed image is addressed.

Another advantage of the AwPCSD algorithm is that there are less number of sensitive hyper-parameters required to be calibrated. In the AwPCSD algorithm, the main structure of the PCSD algorithm is employed. The step size of the steepest descent process is adaptively adjusted according to the difference in the projection domain from the SART step in the current iteration. By doing so, the calibration of some sensitive hyper-parameters can be avoided. These two features of the AwPCSD algorithm in combination pose a great advantage of the AwPCSD algorithm in a sense that the edges of the reconstructed image are better preserved with less number of hyper-parameters to be tuned.

The next contribution of the thesis is to investigate the effects that the reconstruction may have from different settings of the hyper-parameters in the TV regularised algorithms. The

values of hyper-parameters play an important role in the reconstruction performance of the algorithms. A careful selection of the optimal hyper-parameter setting for a given data is one of the main reasons behind the complications of implementing the iterative algorithms, especially for the TV regularised algorithms. The manual tuning of the hyper-parameters is a tedious and time-consuming process. Chapter 5 presents a comprehensive valuation of hyper-parameters to examine the sensitivity that the reconstructed image has to the change in values of hyper-parameters. The aim is to know which hyper-parameters to prioritise when tuning the algorithms to minimise or avoid re-running the algorithms with different values of hyper-parameters. The sensitivity of hyper-parameters in chapter 5 is analysed based on the 4 common TV regularised algorithm including ASD-POCS, AwTV-POCS, PCSD and AwPCSD. By doing so, it is possible to evaluate the performance of the AwPCSD algorithm and observe the difference between the results from the algorithms which minimise the adaptive-weighted TV norm and conventional TV norm.

The results from chapter 5 demonstrate that the choice of hyper-parameters is crucial for the implementation of TV-based regularisation algorithms, especially for the following hyper-parameters: α, α_{red} and r_{max}. Also, the minimisation of adaptive-weighted TV norm is able to preserve the edges of the reconstructed image better and make the algorithms more robust to changes compared to the non-adaptive-weighted algorithms. These two findings confirm the efficacy of the AwPCSD algorithm as this algorithm is able to preserve the edges of the reconstructed image better with less sensitive hyper-parameters to calibrate.

We have seen the evidence from the results so far that the TV-based algorithms only work efficiently if the hyper-parameters in those algorithms are specified optimal values for a given data. The complication of hyper-parameter selection makes it difficult to unlock the real potential of iterative algorithms in practical usage, especially in the medical imaging application. It is then strongly desired to have an automated algorithm to assist in hyper-parameter selection. This topic has become an active area of research in the recent years.

In this thesis, 2 hyper-parameter selection methods for CT reconstruction are developed for this purpose. The AwPCSD algorithm proposed in this thesis is used as a main reconstruction algorithm. Two different techniques, which are successfully used for hyper-parameter selection in different applications, are adapted to identify the best set of hyper-parameters for the AwPCSD algorithm. The AwPCSD algorithm is only implemented here as a representative of the TV regularised algorithm. The hyper-parameter selection methods are generalised to be used with any other reconstruction algorithms with some modifications.

The first hyper-parameter selection algorithm employed the ACO approach, which launches generations of artificial ants that swarm through the AwPCSD algorithm. Each ant represents different configurations of hyper-parameters. One iteration of the AwPCSD algorithm is implemented using each configuration and produces a reconstructed image. A score is given for each image and used to compute a pheromone for each value of hyper-parameter configuration. A higher score leads to stronger pheromone to attract ants in the next generations. Once all generations of ants finish their moves, the image with the highest score in one generation is used as a base image for the next iteration of the algorithm. The same process is repeated in each

iteration of the algorithm until either stopping criterion of the AwPCSD algorithm are met or maximum number of iteration is reached.

The best hyper-parameter configuration suggested by the ACO algorithm produces superior results to those of the non-optimal setting and the CGLS algorithm. Although the results are slightly inferior to the results from the cross-validation method as measured by quantitative metrics, the computational time taken to implement the ACO algorithm is much shorter. The ACO algorithm is also robust against the extra levels of noise added to the projection data. In addition, the optimal hyper-parameter configuration is able to reconstruct a good quality image for other projection data of the same context of imaging.

The second hyper-parameter selection algorithm is developed to overcome drawbacks of the first algorithm. Firstly, the algorithm is aimed to achieve better results than those of the cross-validation method. Secondly, the algorithm is implemented in such a way that the reference image is not required for a comparison during the implementation. This will make the algorithm more applicable to the limited angular real projection data where the exact phantom image is not available and the FDK result from incomplete set of projection data is not a good quality image.

In the second hyper-parameter selection algorithm, the AwPCSD algorithm is also used as a main reconstruction algorithm. The hedge method of Freund and Shapire, which is the online prediction model to solve dynamic allocation problems, is combined with the AwPCSD algorithm to select the best setting of hyper-parameter configuration. The algorithm starts with an initial image reconstructed using a pre-specified number of preliminary projection. The sequential reconstruction of the AwPCSD algorithm is performed for all the hyper-parameter configurations. Then, a forward projection of each reconstructed image is computed and prediction is performed for the next observed projection. The error of the prediction is measured and recorded for all the hyper-parameter configurations. The vector of errors are used to update the probability mass by using the multiplicative rule. Then, the new projection which is the next observed projection is added to the set of projection used to reconstruct an image. The algorithm is moved on to the next iteration and the same process is repeated again until all the available number of projections is used. At the end of the implementation, the best hyper-parameter configuration is determined from the one with the highest probability mass. The main result of this hyper-parameter selection algorithm is that choosing the value of hyper-parameters using the probability mass after a certain number of steps provides a prediction error which is almost as accurate as the prediction error incurred by the best predictor.

The results from the reconstruction using 50 projections and the best hyper-parameters obtained from the Hedge algorithm are compared with the ones from the ACO and the cross-validation algorithms and the reference image. Visually, the results from all 3 algorithms are very similar to each other with no outstanding difference among the algorithms. Also, the reconstructed images look similar to the reference image which suggest that the images are almost as the same quality as the true phantom example. Further comparison using quantitative metrics (i.e. relative errors and UQI) show that the result from the Hedge algorithm is the best among the 3 algorithms, with the lowest relative error and the highest UQI.

It is worth mentioning that the long computational times are only for the training stage of the hyper-parameter selection algorithms. Once the best set of hyper-parameters are found, the implementation of the AwPCSD reconstruction algorithm does not take long time to finish, approximately 2 minutes for the reconstruction of XCAT phantom data using the same testing computer. However, one has to bear in mind that the reconstruction time depends on several factors such number of projection, maximum number of iteration , size of the data and the image to be reconstructed and the computational power of the computer, etc. Although both hyper-parameter selection algorithms presented in this thesis take substantial amount of time in the training stage, it is guaranteed that the best set of hyper-parameters is achieved at the end of the implementation. Thus, the tedious process of manual tuning of hyper-parameter can be avoided. The best set of hyper-parameters of a given data can also be applied to other datasets of similar context of imaging, which saves a lot of time and effort to re-select the hyper-parameters for different data.

The conclusion can be made from the experimental results that two hyper-parameter selection algorithms proposed in this thesis, the ACO and the Hedge algorithms, have been proved to achieve the best set of hyper-parameters for a given data. However, the image quality of the results are slightly different with the Hedge's result is slightly better and computational time is relatively much longer. The requirement for the image quality and the available time frame for the training stage are then the two factors that define which algorithm is more appropriate in which situation. The fact that the result from the Hedge algorithm is quantitatively better than the ACO algorithm suggest that the Hedge algorithm is appropriate for the situation where a more accurate reconstruction is required. However, one has to bear in mind that the accuracy of reconstruction comes at the cost of longer computational time. When the time constraint is more strict, the ACO algorithm is a better alternative since the computational time is drastically shorter than the Hedge algorithm with only slightly inferior result.

Another important point is that the implementation of the ACO algorithm heavily relies on the quality of the reference image as the comparison is always done between the reconstructed image from each hyper-parameter configuration and the reference image in each iteration. The Hedge algorithm does not require the reference image since the errors are measured from the prediction in the projection domain. Due to this fact, the Hedge algorithm is then easily applicable to the case when the reference image is not available such as the projection data from real measurement, specifically in the limited angular scenario where the quality of reconstruction result from the FDK algorithm is not good and cannot be used as a reference image.

One of the reasons behind long computational time for the training stage of two hyper-parameter selection algorithms is because the hyper-parameters being optimised are not all the global parameters. The two hyper-parameter selection algorithms select the hyper-parameters for the implementation of the AwPCSD reconstruction algorithm, which is based on the two-stage approach as mentioned earlier. In the experiments, the two hyper-parameters being selected were Data-inconsistency-tolerance parameter (ε) and TV sub-iteration number (ng). The ε hyper-parameter is the maximum L_2 norm error to accept image as valid and is used as one of the stopping criterion of the AwPCSD algorithm. This hyper-parameter is a global parameter

that governs the current iteration of the algorithm as a whole. The other hyper-parameter, ng, is the TV sub-iteration number which specifies how many times the TV minimisation process performs in each iteration of the algorithm. The ng hyper-parameter controls the effect of the second stage of the current iteration of the algorithm. The change incurred from specifying the values of ng will affect the first stage of the next iteration of the algorithm. Thus, the ng hyper-parameter is not a global parameter. The computational time of the hyper-parameter selection algorithms would have been drastically reduced if all the hyper-parameters being selected are global parameters. In which case, the value of one hyper-parameter can be fixed while running the matrix of different values of another hyper-parameter. The algorithm can be implemented for the smallest value first and then move on to the greater values based on the results of the previous values. However, the hyper-parameters being dealt with in this thesis are not all global parameters. The implementation of the hyper-parameter selection in the mentioned fashion is then not feasible.

It is also important to discuss about the limitations of the work done in this thesis. The first limitation is regarding the algorithmic parameters, which are differed from one algorithm to another depending on the implementation. For instance, the algorithmic parameters for the ACO algorithm are maximum number of iteration, maximum number of generation of ant colony, number of ants in a colony, evaporation rate and the number of hyper-parameter configurations. For the Hedge algorithm, the algorithmic parameters are pre-specified number of preliminary projections (T) and the number of hyper-parameter configurations. All these algorithmic parameters shape the direction of the algorithms and also strongly affect the results and computational time of the algorithms. Most of these parameters are specified the values in the experiments of this thesis based on the trial-and-errors. Next, the rationale behind the choices of two important algorithmic parameters are discussed.

The first one is the initial values of hyper-parameters to be selected by both of the presented hyper-parameter selection algorithms. The ranges of values for the 2 hyper-parameters are chosen such that they represent all possible levels that could affect the reconstruction results. The total number of hyper-parameter configuration is 150, which is a substantial number and causes a long computational time. However, it can be assured that all possible searching space of the hyper-parameters has already been covered.

The next algorithmic parameter is the maximum number of iteration, which is fixed to reflect the time constraint of the reconstruction. In all the experiments of this thesis, the maximum number of iteration was specified as 50 from trial-and-error because this is a reasonable point to observe the reconstruction results. If the maximum number of iteration is specified at a higher number than 50, it is possible that the quality of the reconstructed image can be improved but unfortunately comes at the cost of longer computational time. Thus, the maximum number of iteration for any algorithms is set to 50 such that the time constraint remains constant. However, this might also mean that the algorithms would not reach the small level of error when small values of ε are specified since the algorithm is stopped due to the maximum number of iteration is reached first. The algorithms will aim to reach those small values of ε and the quality of the reconstructed image is evaluated based on this attempt, in combination with the performance

as affected by the other hyper-parameters.

8.1 Future Work

The possible future work of this thesis can be expanded in several aspects.

The first aspect regards the application of the hyper-parameter selection algorithm with real clinical data, especially in a limited angular scenario. In this case, it is obvious that there is no exact phantom available and obtaining a good quality reference image from the FDK algorithm is challenging since the FDK algorithm does not perform well with insufficient number of projection data. The hyper-parameter selection algorithm using the Hedge algorithm would be most suitable for the situation as it does not require the reference image for a comparison during the implementation. The algorithm can be applied to any set of data straight away.

This idea leads to the next possible future work of this thesis in the situation where multiple clinical data sets are available from different patients in the same context of the CT scan such as lung or brain projection data. The hyper-parameter selection algorithms can be applied for the training purpose using each data set. The best hyper-parameter configuration from each data set is learned and the average values over the set of projection data can be found. If sufficient numbers of data are observed, it is promising that the average values from the set of hyper-parameters found from the experiments can be applied to the other data set in a similar context. Although the computational time of the Hedge algorithm for the training is quite long, the fact that the set of hyper-parameters from the training of different data sets can be readily applied to any data without having to undergo the training again can considerably save a lot of time and resource for the reconstruction of CT imaging.

The next challenging extension of this thesis is to apply the hyper-parameter selection algorithms with other different reconstruction algorithms. The two hyper-parameter selection algorithms proposed in this thesis are based on the AwPCSD reconstruction algorithm. However, the implementation is generalised to be used to select hyper-parameters for any other reconstruction algorithms. Some modifications will be required as different algorithms require different numbers of hyper-parameters to implement but the concept of the hyper-parameter selection using ACO and hedge algorithms should still be practicable to other scenarios as well.

In addition, the proposed 2 hyper-parameter selection algorithms can be implemented with a more discrete values of hyper-parameters, other than the ones used in the experiments in chapter 6 and 7, in case that the behaviour of different values of hyper-parameters are required to be observed. However, one has to bear in mind that the number of initial hyper-parameters also affects the computational time of the algorithms.

The next aspect of future work regards the choices of image quality metrics used throughout the experiments in this thesis. For examples, the CC is used to compute the score for the ACO hyper-parameter selection algorithm and the RMSE error is measured in the cross-validation method, as well as the Hedge algorithm. The results in the experiments for these two hyper-parameter selection algorithms are quantitatively measured using the UQI metric. These metrics are chosen only to represent any other metrics that could be used to measure either the similarity

or discrepancy between the results and the reference. The next phase of the thesis could be extended to experiment different image quality metrics in the implementation and evaluation of all the algorithms.

The reconstruction algorithms in this thesis are developed based on the algorithms available in the TIGRE toolbox with the projection and backprojection operators that are optimised for GPU computing using CUDA. The next phase of the TIGRE toolbox will be implemented based on the multi-GPU, which will make the reconstruction algorithms able to tackle larger scale of problems and the computational time will also be reduced.

Bibliography

[1] P. Alaei and E. Spezi. Imaging dose from cone beam computed tomography in radiation therapy. *Physica Medica*, 31(7):647–658, Nov. 2015.

[2] P. Alaei, E. Spezi, and M. Reynolds. Dose calculation and treatment plan optimization including imaging dose from kilovoltage cone beam computed tomography. *Acta Oncologica*, 53(6):839–844, Jan. 2014.

[3] D. Amelunxen, M. Lotz, M. B. McCoy, and J. A. Tropp. Living on the edge: phase transitions in convex programs with random data. *Information and Inference*, 3(3):224–294, June 2014.

[4] A. H. Andersen and A. C. Kak. Simultaneous algebraic reconstruction technique (SART): A superior implementation of the ART algorithm. *Ultrasonic Imaging*, 6(1):81–94, Jan. 1984.

[5] S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4(0):40–79, 2010.

[6] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[7] M. Beister, D. Kolditz, and W. A. Kalender. Iterative reconstruction methods in x-ray CT. *Physica Medica*, 28(2):94–108, Apr. 2012.

[8] M. Bertalmío, V. Caselles, B. Rougé, and A. Solé. Tv based image restoration with local constraints. *Journal of scientific computing*, 19(1-3):95–122, 2003.

[9] A. Biguri, M. Dosanjh, S. Hancock, and M. Soleimani. TIGRE: a MATLAB-GPU toolbox for CBCT image reconstruction. *Biomedical Physics & Engineering Express*, 2(5):055010, Sept. 2016.

[10] Å. Bjorck. *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics, Jan. 1996.

[11] S. Boyd and L. Vandenberghe. *Convex Optimization, With Corrections 2008*. Cambridge University Press, 2004.

[12] D. J. Brenner and E. J. Hall. Computed tomography — an increasing source of radiation exposure. *New England Journal of Medicine*, 357(22):2277–2284, Nov. 2007.

[13] E. Candes and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies?

[14] E. Candes and T. Tao. Decoding by linear programming. *arXiv preprint math/0502327*, 2005.

[15] E. Candes and M. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, Mar. 2008.

[16] E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis?

[17] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

[18] Y. Censor, R. Davidi, G. T. Herman, R. W. Schulte, and L. Tetruashvili. Projected subgradient minimization versus superiorization. *Journal of Optimization Theory and Applications*, 160(3):730–747, Sept. 2013.

[19] Y. Censor and T. Elfving. Block-iterative algorithms with diagonally scaled oblique projections for the linear feasibility problem. *SIAM Journal on Matrix Analysis and Applications*, 24(1):40–58, Jan. 2002.

[20] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97, 2004.

[21] T. F. Chan and C.-K. Wong. Total variation blind deconvolution. *IEEE transactions on Image Processing*, 7(3):370–375, 1998.

[22] G.-H. Chen, J. Tang, and S. Leng. Prior image constrained compressed sensing (PICCS): A method to accurately reconstruct dynamic CT images from highly undersampled projection data sets. *Medical Physics*, 35(2):660–663, Jan. 2008.

[23] Z. Chen, H. Qi, S. Wu, Y. Xu, and L. Zhou. Few-view CT reconstruction via a novel non-local means algorithm. *Physica Medica*, 32(10):1276–1283, Oct. 2016.

[24] K. Choi, J. Wang, L. Zhu, T.-S. Suh, S. Boyd, and L. Xing. Compressed sensing based cone-beam computed tomography reconstruction with a first-order methoda). *Medical Physics*, 37(9):5113–5125, Aug. 2010.

[25] S. Chretien. An alternating l_1 approach to the compressed sensing problem. *IEEE Signal Processing Letters*, 17(2):181–184, Feb. 2010.

[26] S. Chretien, A. Gibberd, and S. Roy. Hedging hyperparameter selection for basis pursuit. *arXiv preprint arXiv:1805.01870*, 2018.

[27] S. B. Coban. Sophiabeads dataset project codes, 2015.

[28] S. B. Coban and S. A. McDonald. Sophiabeads dataset project, 2015.

[29] P. Combettes and J.-C. Pesquet. Image restoration subject to a total variation constraint. *IEEE Transactions on Image Processing*, 13(9):1213–1222, Sept. 2004.

[30] M. Das, A. H. Mahnken, G. MÃ¼hlenbruch, A. Stargardt, C. Weiß, D.-A. Sennst, T. G. Flohr, R. W. GÃ¼nther, and J. E. Wildberger. Individually adapted examination protocols for reduction of radiation exposure for 16-MDCT chest examinations. *American Journal of Roentgenology*, 184(5):1437–1443, May 2005.

[31] M. E. Davison. The ill-conditioned nature of the limited angle tomography problem. *SIAM Journal on Applied Mathematics*, 43(2):428–448, 1983.

[32] P. Dawkins. Paul's online math notes, 2015.

[33] J. Deng, Z. Chen, K. B. Roberts, and R. Nath. Kilovoltage imaging doses in the radiotherapy of pediatric cancer patients. *International Journal of Radiation Oncology Biology Physics*, 82(5):1680–1688, Apr. 2012.

[34] G. X. Ding and C. W. Coffey. Radiation dose from kilovoltage cone beam computed tomography in an image-guided radiotherapy procedure. *International Journal of Radiation Oncology Biology Physics*, 73(2):610–617, Feb. 2009.

[35] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, Apr. 2006.

[36] D. Donoho and J. Tanner. Counting faces of randomly projected polytopes when the projection radically lowers dimension. *Journal of the American Mathematical Society*, 22(1):1–53, 2009.

[37] D. Donoho and J. Tanner. Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1906):4273–4293, Nov. 2009.

[38] D. L. Donoho and J. Tanner. Sparse nonnegative solution of underdetermined linear equations by linear programming. *Proceedings of the National Academy of Sciences*, 102(27):9446–9451, 2005.

[39] D. L. Donoho and J. Tanner. Exponential bounds implying construction of compressed sensing matrices, error-correcting codes, and neighborly polytopes by random sampling. *IEEE Transactions on Information Theory*, 56(4):2002–2016, Apr. 2010.

[40] M. Dorigo, G. D. Caro, and L. M. Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172, Apr. 1999.

[41] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: An autocatalytic optimizing process. 1991.

[42] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1):29–41, 1996.

[43] A. J. Einstein, M. J. Henzlova, and S. Rajagopalan. Estimating risk of cancer associated with radiation exposure from 64-slice computed tomography coronary angiography. *JAMA*, 298(3):317, July 2007.

[44] L. Feldkamp, L. Davis, and J. Kress. Practical cone-beam algorithm. *J.Opt.Soc.Am.A.*, 1(6):612–619, 1984.

[45] L. A. Feldkamp, L. C. Davis, and J. W. Kress. Practical cone-beam algorithm. *Journal of the Optical Society of America A*, 1(6):612, June 1984.

[46] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, Aug. 1997.

[47] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

[48] P. Gilbert. Iterative methods for the three-dimensional reconstruction of an object from projections. *Journal of Theoretical Biology*, 36(1):105–117, July 1972.

[49] R. Glowinski and A. Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 9(R2):41–76, 1975.

[50] T. Goldstein and S. Osher. The split bregman method for l1-regularized problems. *SIAM journal on imaging sciences*, 2(2):323–343, 2009.

[51] G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

[52] R. Gordon, R. Bender, and G. T. Herman. Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and x-ray photography. *Journal of Theoretical Biology*, 29(3):471–481, Dec. 1970.

[53] W. Guo and W. Yin. Edgecs: Edge guided compressive sensing reconstruction. In *Visual Communications and Image Processing 2010*, volume 7744, page 77440L. International Society for Optics and Photonics, 2010.

[54] P. C. Hansen. Analysis of discrete ill-posed problems by means of the l-curve. *SIAM Review*, 34(4):561–580, Dec. 1992.

[55] P. C. Hansen. Regularization tools: A matlab package for analysis and solution of discrete ill-posed problems. *Numerical algorithms*, 6(1):1–35, 1994.

[56] P. C. Hansen and M. Saxild-Hansen. AIR tools — a MATLAB package of algebraic iterative reconstruction methods. *Journal of Computational and Applied Mathematics*, 236(8):2167–2178, Feb. 2012.

[57] E. Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[58] G. T. Herman. *Fundamentals of computerized tomography: image reconstruction from projections*. Springer Science & Business Media, 2009.

[59] J. Hsieh. *Computed Tomography: Principles, Design, Artifacts, and Recent Advances*, volume 114. SPIE Press, 2003.

[60] H. M. Hudson and R. S. Larkin. Accelerated image reconstruction using ordered subsets of projection data. *IEEE transactions on medical imaging*, 13(4):601–609, 1994.

[61] F. Jacobs, E. Sundermann, B. De Sutter, M. Christiaens, and I. Lemahieu. A fast algorithm to calculate the exact radiological path through a pixel or voxel space. *Journal of computing and information technology*, 6(1):89–94, 1998.

[62] U. Je, M. Lee, H. Cho, D. Hong, Y. Park, C. Park, H. Cho, S. Choi, and T. Woo. Simulation and experimental studies of three-dimensional (3d) image reconstruction from insufficient sampling data based on compressed-sensing theory for potential applications to dental cone-beam CT. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 784:550–556, June 2015.

[63] A. Jerri. The shannon sampling theorem—its various extensions and applications: A tutorial review. *Proceedings of the IEEE*, 65(11):1565–1596, 1977.

[64] X. Jia, B. Dong, Y. Lou, and S. B. Jiang. GPU-based iterative cone-beam CT reconstruction using tight frame regularization. *Physics in Medicine and Biology*, 56(13):3787–3807, May 2011.

[65] X. Jia, Y. Lou, R. Li, W. Y. Song, and S. B. Jiang. GPU-based fast cone beam CT reconstruction from undersampled and noisy projection data via total variation. *Medical Physics*, 37(4):1757–1760, Mar. 2010.

[66] X. Jia, Z. Tian, Y. Lou, J.-J. Sonke, and S. B. Jiang. Four-dimensional cone beam ct reconstruction and enhancement using a temporal nonlocal means method. *Medical physics*, 39(9):5592–5602, 2012.

[67] B. Jones and D. A. Morgan. Chapter 4 radiotherapy fractionation. In *Radiobiological Modelling in Radiation Oncology*, pages 51–78. The British Institute of Radiology, Jan. 2007.

[68] J. S. Jørgensen, S. B. Coban, W. R. Lionheart, and P. J. Withers. Effect of sparsity and exposure on total variation regularized x-ray tomography from few projections. In *4th International Conference on Image Formation in X-Ray Computed Tomography*, pages 279–282, 2016.

[69] J. S. Jorgensen and E. Y. Sidky. How little data is enough? phase-diagram analysis of sparsity-regularized x-ray computed tomography. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 373(2043):20140387–20140387, May 2015.

[70] S. Kaczmarz. Angenäherte Auflösung von Systemen linearer Gleichungen. *Bulletin International de l'Académie Polonaise des Sciences et des Lettres*, 35:355â€“357, 1937.

[71] A. C. Kak and M. Slaney. *Principles of Computerized Tomographic Imaging*. I.E.E.E.Press, 1988.

[72] A. C. Kak and M. Slaney. *Principles of Computerized Tomographic Imaging (Classics in Applied Mathematics)*. Society for Industrial and Applied Mathematics, 2001.

[73] M. K. Kalra, C. Wittram, M. M. Maher, A. Sharma, G. B. Avinash, K. Karau, T. L. Toth, E. Halpern, S. Saini, and J.-A. Shepard. Can noise reduction filters improve low-radiation-dose chest CT images? pilot study. *Radiology*, 228(1):257–264, July 2003.

[74] K. Kim. 3d cone beam ct (cbct) projection backprojection fdk, iterative reconstruction matlab examples. *Mathworks. March*, 10, 2012.

[75] K. Lange, R. Carson, et al. Em reconstruction algorithms for emission and transmission tomography. *J Comput Assist Tomogr*, 8(2):306–16, 1984.

[76] K. A. Langmack. Portal imaging. *The British Journal of Radiology*, 74(885):789–804, Sept. 2001.

[77] J. Liesen and P. Tichý. Convergence analysis of krylov subspace methods. *GAMM-Mitteilungen*, 27(2):153–173, Dec. 2004.

[78] L. Liu, W. Lin, and M. Jin. Reconstruction of sparse-view x-ray computed tomography using adaptive iterative algorithms. *Computers in Biology and Medicine*, 56:97–106, Jan. 2015.

[79] L. Liu, Z. Yin, and X. Ma. Nonparametric optimization of constrained total variation for tomography reconstruction. *Computers in Biology and Medicine*, 43(12):2163–2176, Dec. 2013.

[80] Y. Liu, J. Ma, Y. Fan, and Z. Liang. Adaptive-weighted total variation minimization for sparse data toward low-dose x-ray computed tomography image reconstruction. *Physics in Medicine and Biology*, 57(23):7923–7956, Nov. 2012.

[81] M. Lohvithee, A. Biguri, and M. Soleimani. Parameter selection in limited data cone-beam CT reconstruction using edge-preserving total variation algorithms. *Physics in Medicine & Biology*, 62(24):9295–9321, Nov. 2017.

[82] A. K. Louis. Incomplete data problems in x-ray computerized tomography. *Numerische Mathematik*, 48(3):251–262, May 1986.

[83] X. Lu, Y. Sun, and Y. Yuan. Optimization for limited angle tomography in medical image processing. *Pattern Recognition*, 44(10-11):2427–2435, Oct. 2011.

[84] J. Ma, Z. Liang, Y. Fan, Y. Liu, J. Huang, W. Chen, and H. Lu. Variance analysis of x-ray CT sinograms in the presence of electronic noise background. *Medical Physics*, 39(7Part1):4051–4065, June 2012.

[85] S. H. Manglos, G. M. Gagne, A. Krol, F. D. Thomas, and R. Narayanaswamy. Transmission maximum-likelihood reconstruction with ordered subsets for cone beam CT. *Physics in Medicine and Biology*, 40(7):1225–1241, July 1995.

[86] M. T. McCann, M. Nilchian, M. Stampanoni, and M. Unser. Fast 3d reconstruction method for differential phase contrast x-ray CT. *Optics Express*, 24(13):14564, June 2016.

[87] C. H. McCollough, A. N. Primak, N. Braun, J. Kofler, L. Yu, and J. Christner. Strategies for reducing radiation dose in CT. *Radiologic Clinics of North America*, 47(1):27–40, Jan. 2009.

[88] M. G. McGaffin and J. A. Fessler. Alternating dual updates algorithm for x-ray CT reconstruction on the GPU. *IEEE Transactions on Computational Imaging*, 1(3):186–199, Sept. 2015.

[89] N. Meinshausen and P. BÃ¼hlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, June 2006.

[90] B. Meng, L. Xing, B. Han, A. Koong, D. Chang, J. Cheng, and R. Li. Cone beam CT imaging with limited angle of projections and prior knowledge for volumetric verification of non-coplanar beam radiation therapy: a proof of concept study. *Physics in Medicine and Biology*, 58(21):7777–7789, Oct. 2013.

[91] M. Miften, O. Gayou, B. Reitz, R. Fuhrer, B. Leicher, and D. S. Parda. IMRT planning and delivery incorporating daily dose from mega-voltage cone-beam computed tomography imaging. *Medical Physics*, 34(10):3760–3767, Sept. 2007.

[92] K. Mueller. *Fast and accurate three-dimensional reconstrution from cone-beam projection data using algebraic methods*. PhD thesis, The Ohio State University, 1998.

[93] K. Mueller, R. Yagel, and J. Wheller. Anti-aliased three-dimensional cone-beam reconstruction of low-contrast objects with algebraic methods. *IEEE Transactions on Medical Imaging*, 18(6):519–537, June 1999.

[94] K. Mueller, R. Yagel, and J. Wheller. Fast implementations of algebraic methods for three-dimensional reconstruction from cone-beam data. *IEEE Transactions on Medical Imaging*, 18(6):538–548, June 1999.

[95] T. H. Mulkens, P. Bellinck, M. Baeyaert, D. Ghysen, X. V. Dijck, E. Mussen, C. Venstermans, and J.-L. Termote. Use of an automatic exposure control mechanism for dose optimization in multi–detector row CT examinations: Clinical evaluation. *Radiology*, 237(1):213–223, Oct. 2005.

[96] F. Natterer. *The mathematics of computerized tomography*, volume 32. Siam, 1986.

[97] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[98] J. Ni, X. Li, T. He, and G. Wang. Review of parallel computing techniques for computed tomography image reconstruction. *Current Medical Imaging Reviews*, 2(4):405–414, 2006.

[99] J. Nocedal and S. Wright. *Numerical Optimization (Springer Series in Operations Research and Financial Engineering)*. Springer, 2000.

[100] C. C. Paige and M. A. Saunders. Lsqr: An algorithm for sparse linear equations and sparse least squares. *ACM transactions on mathematical software*, 8(1):43–71, 1982.

[101] X. Pan, E. Y. Sidky, and M. Vannier. Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction? *Inverse Problems*, 25(12):123009, Dec. 2009.

[102] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, July 1990.

[103] B. T. Poljak. A general method for solving extremum problems. *Sov. Math. Dokl.*, 8(3):593–597, 1967.

[104] B. T. Polyak. Minimization of unsmooth functionals. *USSR Computational Mathematics and Mathematical Physics*, 9(3):14–29, 1969.

[105] F. Pontana, A. Duhamel, J. Pagniez, T. Flohr, J.-B. Faivre, A.-L. Hachulla, J. Remy, and M. Remy-Jardin. Chest computed tomography using iterative reconstruction vs filtered back projection (part 2): image quality of low-dose CT examinations in 80 patients. *European Radiology*, 21(3):636–643, Nov. 2010.

[106] F. Pontana, J. Pagniez, T. Flohr, J.-B. Faivre, A. Duhamel, J. Remy, and M. Remy-Jardin. Chest computed tomography using iterative reconstruction vs filtered back projection (part 1): evaluation of image noise reduction in 32 patients. *European Radiology*, 21(3):627–635, Nov. 2010.

[107] S. Ramani and J. A. Fessler. Convergent iterative ct reconstruction with sparsity-based regularization. In *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med*, pages 302–5, 2011.

[108] S. Ramani, Z. Liu, J. Rosen, J. Nielsen, and J. A. Fessler. Regularization parameter selection for nonlinear iterative image restoration and MRI reconstruction using GCV and SURE-based methods. *IEEE Transactions on Image Processing*, 21(8):3659–3672, Aug. 2012.

[109] S. Rit, M. V. Oliva, S. Brousmiche, R. Labarbe, D. Sarrut, and G. C. Sharp. The reconstruction toolkit (RTK), an open-source cone-beam CT reconstruction toolkit based on the insight toolkit (ITK). *Journal of Physics: Conference Series*, 489:012079, Mar. 2014.

[110] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, Nov. 1992.

[111] A. P. Ruszczyński and A. Ruszczynski. *Nonlinear optimization*, volume 13. Princeton university press, 2006.

[112] V. Sarkar, C. Shi, P. Rassiah-Szegedi, A. Diaz, T. Eng, and N. Papanikolaou. The effect of a limited number of projections and reconstruction algorithms on the image quality of megavoltage digital tomosynthesis. *Journal of applied clinical medical physics*, 10(3):155–172, 2009.

[113] K. Sauer and B. Liu. Nonstationary filtering of transmission tomograms in high photon counting noise. *IEEE Transactions on Medical Imaging*, 10(3):445–452, 1991.

[114] W. C. Scarfe and A. G. Farman. What is cone-beam CT and how does it work? *Dental Clinics of North America*, 52(4):707–730, Oct. 2008.

[115] W. P. Segars, G. Sturgeon, S. Mendonca, J. Grimes, and B. M. W. Tsui. 4d XCAT phantom for multimodality imaging research. *Medical Physics*, 37(9):4902–4915, Aug. 2010.

[116] C. C. Shaw, editor. *Cone Beam Computed Tomography (Imaging in Medical Diagnosis and Therapy)*. CRC Press, 2014.

[117] M. V. Shcherbakov, A. Brebels, N. L. Shcherbakova, A. P. Tyukov, T. A. Janovsky, and V. A. Kamaev. A survey of forecast error measures. *World Applied Sciences Journal*, 24(24):171–176, 2013.

[118] N. Z. Shor. *Minimization methods for non-differentiable functions*, volume 3. Springer Science & Business Media, 2012.

[119] R. L. Siddon. Fast calculation of the exact radiological path for a three-dimensional CT array. *Medical Physics*, 12(2):252–255, Mar. 1985.

[120] E. Y. Sidky, J. H. Jørgensen, and X. Pan. Convex optimization problem prototyping for image reconstruction in computed tomography with the chambolle–pock algorithm. *Physics in Medicine and Biology*, 57(10):3065–3091, Apr. 2012.

[121] E. Y. Sidky, C.-M. Kao, and X. Pan. Accurate image reconstruction from few-views and limited-angle data in divergent-beam ct. *Journal of X-Ray Science and Technology*, 14(2):119–139, 2006.

[122] E. Y. Sidky and X. Pan. Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. *Physics in Medicine and Biology*, 53(17):4777–4807, Aug. 2008.

[123] D. Tack, V. D. Maertelaer, and P. A. Gevenois. Dose reduction in multidetector CT using attenuation-based online tube current modulation. *American Journal of Roentgenology*, 181(2):331–334, Aug. 2003.

[124] M. Tao, J. Yang, and B. He. Alternating direction algorithms for total variation deconvolution in image reconstruction. *TR0918, Department of Mathematics, Nanjing University*, 2009.

[125] Z. Tian, X. Jia, K. Yuan, T. Pan, and S. B. Jiang. Low-dose CT reconstruction via edge-preserving total variation regularization. *Physics in Medicine and Biology*, 56(18):5949–5967, Aug. 2011.

[126] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[127] W. van Aarle, W. J. Palenstijn, J. D. Beenhouwer, T. Altantzis, S. Bals, K. J. Batenburg, and J. Sijbers. The ASTRA toolbox: A platform for advanced algorithm development in electron tomography. *Ultramicroscopy*, 157:35–47, Oct. 2015.

[128] W. S. Van Hemelryck Tessa, G. Maggie, B. K. Joost, and S. Jan. *The implementation of iterative reconstruction algorithms in MATLAB*. PhD thesis, Masterâ€™s thesis, Department of industrial sciences and technology, University college of Antwerp, Belgium, 2007.

[129] C. R. Vogel and M. E. Oman. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1):227–238, 1996.

[130] G. Wang and M. Jiang. Ordered-subset simultaneous algebraic reconstruction techniques (os-sart). *Journal of X-ray Science and Technology*, 12(3):169–177, 2004.

[131] J. Wang, T. Li, Z. Liang, and L. Xing. Dose reduction for kilovotage cone-beam computed tomography in radiation therapy. *Physics in Medicine and Biology*, 53(11):2897–2909, May 2008.

[132] Y. Wang, J. Yang, W. Yin, and Y. Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, Jan. 2008.

[133] Z. Wang and A. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84, Mar. 2002.

[134] Z. Wang and A. C. Bovik. A universal image quality index. *IEEE signal processing letters*, 9(3):81–84, 2002.

[135] J. Xu and B. Tsui. Electronic noise modeling in statistical iterative reconstruction. *IEEE Transactions on Image Processing*, 18(6):1228–1238, June 2009.

[136] W. Xu and K. Mueller. Learning effective parameter settings for iterative ct reconstruction algorithms. In *Workshop on High Performance Image Reconstruction (HPIR)*. Citeseer, 2009.

[137] W. Xu and K. Mueller. Parameter space visualizer: an interactive parameter selection interface for iterative CT reconstruction algorithms. In *Medical Imaging 2010: Visualization, Image-Guided Procedures, and Modeling*. SPIE, Mar. 2010.

[138] W. Xu and K. Mueller. Using GPUs to learn effective parameter settings for GPU-accelerated iterative CT reconstruction algorithms. In *GPU Computing Gems Emerald Edition*, pages 693–708. Elsevier, 2011.

[139] H. Xue, L. Zhang, Z. Cheng, Y. Xing, and Y. Xiao. An improved tv minimization algorithm for incomplete data problem in computer tomography. In *IEEE Nuclear Science Symposuim & Medical Imaging Conference*, pages 2621–2624. IEEE, 2010.

[140] Z. Xue, L. Zhang, and J. Pan. Fast real-time image reconstruction with helical cone-beam ART. In *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*. IEEE, Oct. 2010.

[141] G. Yang. *Numerical Approaches for Solving the Combined Reconstruction and Registration of Digital Breast Tomosynthesis*. PhD thesis, UCL (University College London), 2012.

[142] J. Yang, W. Yin, Y. Zhang, and Y. Wang. A fast algorithm for edge-preserving variational multichannel image restoration. *SIAM Journal on Imaging Sciences*, 2(2):569–592, Jan. 2009.

[143] G. Zeng. *Medical Image Reconstruction: A Conceptual Tutorial*. Springer, 2010.

[144] H. Zhang, L. Ouyang, J. Huang, J. Ma, W. Chen, and J. Wang. Few-view cone-beam ct reconstruction with deformed prior image. *Medical physics*, 41(12), 2014.

[145] T. Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research*, 11(Mar):1081–1107, 2010.

[146] Z. Zhao, G. J. Gang, and J. H. Siewerdsen. Noise, sampling, and the number of projections in cone-beam CT with a flat-panel detector. *Medical Physics*, 41(6Part1):061909, May 2014.

[147] Z. Zheng, E. Papenhausen, and K. Mueller. Searching effective parameters for low-dose ct reconstruction by ant colony optimization. In *Int. Conf. Image Formation in X-Ray Computed Tomography*, pages 182–185, 2012.

[148] Z. Zheng, E. Papenhausen, and K. Mueller. DQS advisor: a visual interface and knowledge-based system to balance dose, quality, and reconstruction speed in iterative CT reconstruction with application to NLM-regularization. *Physics in Medicine and Biology*, 58(21):7857–7873, Oct. 2013.

[149] W. Zhuang, S. Gopal, and T. Hebert. Numerical evaluation of methods for computing tomographic projections. *IEEE Transactions on Nuclear Science*, 41(4):1660–1665, 1994.